Métagénomique comparative de novo à grande échelle

Gaëtan Benoit

ANNÉE 2017

UNIVERSITÉ DE RENNES 1

UNIVERSITE BRETAGNE LOIRE

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

École doctorale MATHSTIC

présentée par

Gaëtan BENOIT

préparée à l'unité de recherche IRISA – UMR6074
Institut de Recherche en Informatique et Système Aléatoires
ISTIC

Métagénomique comparative *de novo* à grande échelle

Thèse à soutenir le 29/11/2017

devant le jury composé de :

Éric COISSAC
Maître de conférence, LECA / *Rapporteur*
Cédric NOTREDAME
Directeur de recherche, CRG / *Rapporteur*
Loïs MAIGNIEN
Maître de conférence, LM2E / *Examinateur*
Malika AINOUCHE
Professeur, Université de Rennes 1 / *Examinatrice*
Dominique LAVENIER
Directeur de recherche, IRISA / *Directeur de thèse*
Claire LEMAITRE
Chargé de recherche, IRISA / *Co-directrice de thèse*

Résumé

La métagénomique vise à étudier le contenu génomique d'un échantillon extrait d'un milieu naturel. Parmi les analyses de données métagénomiques, la métagénomique comparative a pour objectif d'estimer la similarité entre deux ou plusieurs environnements d'un point de vue génomique. L'approche traditionnelle compare les échantillons sur la base des espèces identifiées. Cependant, cette méthode est biaisée par l'incomplétude des bases de données de références.

La métagénomique comparative est dite *de novo* lorsque les échantillons sont comparés sans connaissances *a priori*. La similarité est alors estimée en comptant le nombre de séquences d'ADN similaires entre les jeux de données. Un projet métagénomique génère typiquement des centaines de jeux de données. Chaque jeu contient des dizaines de millions de courtes séquences d'ADN de 100 à 200 nucléotides (appelées lectures). Dans le contexte du début de cette thèse, il aurait fallu des années pour comparer une telle masse de données avec les méthodes usuelles. Cette thèse présente des approches *de novo* pour calculer très rapidement la similarité entre de nombreux jeux de données.

Les travaux que nous proposons se basent sur le k-mer (mot de taille k) comme unité de comparaison des métagénomes. La méthode principale développée pendant cette thèse, nommée Simka, calcule de nombreuses mesures de similarité en remplacement les comptages d'espèces classiquement utilisés par des comptages de grands k-mers ($k > 21$). Simka passe à l'échelle sur les projets métagénomiques actuels grâce à un nouvelle stratégie pour compter les k-mers de nombreux jeux de données en parallèle.

Les expériences sur les données du projet Human Microbiome Projet et Tara Oceans montrent que les distances calculées par Simka sont bien corrélées avec les distances basées sur des comptages d'espèces ou d'OTUs. Simka a traité ces projets (plus de 30 milliards de lectures réparties dans des centaines de jeux) en quelques heures. C'est actuellement le seul outil à passer à l'échelle sur une telle quantité de données, tout en étant complet du point de vue des résultats de comparaisons.

Table des matières

1 Introduction **5**
1.1 Génomique et bioinformatique des séquences 5
1.1.1 Séquençage de première génération et génétique 6
1.1.2 Séquençage haut-débit 7
1.1.3 Assemblage de données génomiques 7
1.1.4 Conclusion 8
1.2 Métagénomique 9
1.2.1 Questions et données 9
1.2.2 Analyse de la diversité taxonomique 10
1.2.2.1 Assignation taxonomique 11
1.2.2.2 Détection d'OTUs 12
1.2.2.3 Assemblage métagénomique 13
1.2.2.4 Binning 14
1.2.3 Analyse de la diversité fonctionnelle 16
1.2.4 Métagénomique comparative 17
1.2.5 Applications 17
1.3 Conclusion 19
1.4 Organisation du manuscrit 21

2 État de l'art de la métagénomique comparative *de novo* **23**
2.1 Objectifs et challenges 23
2.2 Comparaison de métagénomes basées sur des comparaisons de lectures 25
2.2.1 Avec alignement de séquences 25
2.2.2 Sans alignement de séquences 26
2.3 Comparaison de métagénomes basées sur des comparaisons de *k*-mers 29
2.3.1 Comparaison basée sur la présence-absence des *k*-mers 30
2.3.2 Comparaison basée sur le comptage des *k*-mers 31
2.3.2.1 Comptage des *k*-mers 32
2.3.2.2 Comparaison des spectres de *k*-mers 34
2.4 Conclusion 37

3 Simka : nouvelle méthode de métagénomique comparative *de novo* à grande échelle basée sur des *k*-mers **39**
3.1 Stratégie 39
3.2 Comptage de *k*-mers multi-jeux 41
3.2.1 Tri-comptage 42
3.2.2 Fusion des comptages 42
3.2.3 Filtre d'abondance des *k*-mers 43
3.3 Calcul des distances 43

3.3.1 Calcul de la distance de Bray-Curtis 44
3.3.2 Autres distances 45
3.3.3 Distances simples et distances complexes 46
3.4 Implémentation 48
3.5 Évaluation des performances 50
3.5.1 Performances sur de petits jeux de lectures 50
3.5.2 Performances sur le projet HMP entier 52
3.5.3 Impact de la taille des *k*-mers 53
3.5.4 Scalabilité de SIMKA 53
3.6 Conclusion 55

4 Évaluation de la qualité des distances calculées par Simka 57
4.1 Évaluation des distances 57
4.1.1 Corrélation avec des approches *de novo* basées sur des comparaisons de lectures. 57
4.1.2 Corrélation avec des distances taxonomiques sur les données d'intestin. 58
4.1.3 Impact des paramètres de Simka 61
4.1.4 Visualisation de la structure des jeux de lectures du projet HMP 64
4.1.5 Résultats sur un environnement complexe 64
4.2 Application aux données de Tara Oceans 68
4.3 Conclusion 72

5 Approches de sous-échantillonnage de données pour le calcul des distances . 75
5.1 Protocole de sous-échantillonnage et d'évaluation 75
5.2 Sous-échantillonnage au niveau des lectures 78
5.2.1 Erreur d'estimation des distances issues du sous-échantillonnage 78
5.2.2 Corrélation entre les distances attendues et les distances observées 78
5.2.3 Impact sur la classification des échantillons 78
5.2.4 Conclusion 83
5.3 Sous-échantillonnage au niveau des vecteurs d'abondances 84
5.3.1 Erreur d'estimation des distances issues du sous-échantillonnage 84
5.3.2 Conclusions 84
5.4 SimkaMin : nouvelle méthode d'estimation de la distance de Bray-Curtis 84
5.4.1 Méthode 86
5.4.1.1 Sélection des *k*-mers. 86
5.4.1.2 Calcul de la distance de Bray-Curtis. 86
5.4.1.3 Filtrage efficace des *k*-mers vus une seule fois. 87
5.4.2 Évaluation de la distance de Bray-Curtis calculée par SimkaMin 88
5.4.2.1 Erreur d'estimation des distances de SimkaMin 88
5.4.2.2 Comparaison avec l'estimation de la distance de Jaccard de MASH (minhash) 88
5.4.2.3 Impact du filtre d'abondance 88
5.4.3 Performances de SimkaMin 90
5.4.4 Conclusion 90
5.5 Conclusion 91

6 Conclusion et perspectives **93**
6.1 Impact de Simka dans la communauté scientifique 94
6.2 Perspectives 95
6.2.1 Amélioration de la sensibilité des distances basées sur les k-mers 95
6.2.2 Amélioration de la robustesse des distances 95
6.2.3 Séquences similaires entre les jeux de données 95
6.2.4 Mesure de complexité intra-échantillon 96
6.2.5 Requêtage de jeux de données métagénomiques 96

Annexes **99**

Bibliographie **101**

Publications **113**

Liste des tableaux **115**

Table des figures **115**

Chapitre 1

Introduction

Les micro-organismes sont présents dans tous les environnements et accomplissent un rôle vital dans (presque) tous les écosystèmes. De plus, certaines communautés microbiennes vivent en symbiose avec leurs hôtes et influent sur leur fonctionnement et leur santé. Par exemple, l'être humain héberge 10 à 100 fois plus de bactéries qu'il n'a de cellules, et celles-ci exercent un nombre de fonctions bien supérieur à celui de ses propres cellules [1]. Réussir à décrire ces communautés microscopiques, à comprendre leurs fonctions et les facteurs qui peuvent influencer leur composition sont des problèmes fondamentaux de la biologie avec des enjeux énormes en ce qui concerne l'environnement et la médecine. La flore intestinale humaine a par exemple récemment été mise en cause dans le développement de certaines maladies telles que l'obésité, le diabète de type 2 ou la maladie de Crohn [2].

Cette introduction généraliste a pour objectif de présenter la métagénomique, discipline qui explore les communautés d'organismes à travers l'analyse de l'ADN prélevé dans des environnements naturels.

Les fondements de la métagénomique, à savoir la génétique et la génomique, sont tout d'abord présentés. Les différentes techniques d'analyse pour extraire des connaissances des jeux de données métagénomiques sont ensuite abordées. Enfin, après avoir présenté les limitations de chacune de ces techniques d'analyses, la métagénomique comparative *de novo* est mise en avant. Celle-ci a pour objectif d'estimer la similarité entre deux environnements d'un point de vue génomique. Pour cela, leurs séquences d'ADN sont comparées entre elles afin de déterminer leur contenu génomique partagé et spécifique. La difficulté de ce problème provient de la taille démesurée des données à comparer. En effet, les projets métagénomiques actuels, tels que le projet Tara Oceans [3], mettent à disposition des centaines de jeux de données contenant chacun des centaines de millions de séquences d'ADN. Dans le contexte du début de cette thèse, il aurait fallu des années pour comparer une telle masse de données avec les méthodes usuelles. Cette thèse s'attaque à réduire ce temps de traitement et à valider les techniques proposées.

1.1 Génomique et bioinformatique des séquences

Les séquenceurs d'ADN déterminent l'enchainement des nucléotides d'un fragment d'ADN et les numérisent. Le résultat de cette opération est un texte représentant la séquence du fragment d'ADN codée sur quatre lettres A, C, G et T, les quatre nucléotides (ou bases) de l'ADN. La bioinformatique des séquences exploite ces données de séquençage en utilisant l'outil informatique afin de répondre à des problèmes biologiques. Cette section introduit les données sur lesquelles s'appuie cette thèse et présente les concepts et processus fondamentaux de la bioinformatique

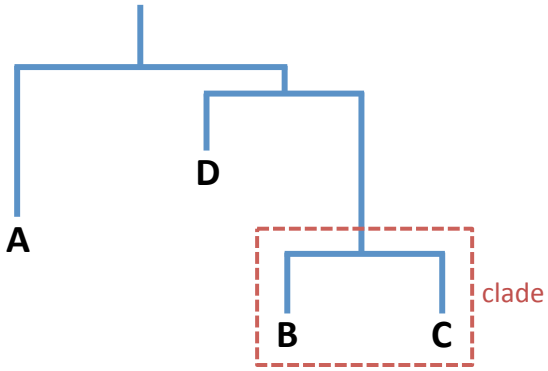

Figure 1.1. Représentation d'une classification phylogénétique. Dendrogramme de quatre espèces A, B, C et D. La taille des branches et le nombre de nœuds entre chaque espèce indique leur distance évolutive. Les espèces B et C appartiennent à un même clade puisqu'il partage un ancêtre commun.

des séquences.

1.1.1 Séquençage de première génération et génétique

Les premières technologies de séquençage automatique de l'ADN, nommées Sanger [4] et Maxam–Gilbert [5], sont apparues indépendamment en 1977. La quantité de données qu'ont mis à disposition ces machines a notamment révolutionné la génétique, discipline qui se consacre à l'étude des gènes entre différents individus ou entre différentes espèces. La même année, en conjonction avec ces technologies, l'utilisation des gènes ribosomiques (ARNr) est proposée pour identifier les êtres vivants et les classifier [6].

Une telle classification conduit à l'obtention d'un arbre phylogénétique, représentatif des liens de parenté entre les organismes étudiés (figure 1.1). Un nœud de l'arbre représente l'ancêtre commun de ses descendants. Plus la longueur de branche entre deux espèces est grande, plus elles sont éloignées. On appelle clade un regroupement d'organismes partageant un ancêtre commun. Les clades ont tendance à remplacer peu à peu la notion de taxon qui représente des organismes ou groupes d'organismes partageant des caractères en commun. Historiquement, les taxons ont cependant été définis sur des caractères morphologiques. Il est donc possible que des espèces d'un même taxon soit en fait éloignées génétiquement et appartiennent à des clades différents.

Pour obtenir une telle classification génétique, les séquences d'ADN sont comparées pour déterminer leur degré de similarité. L'idée consiste à faire correspondre au mieux les deux séquences tout en minimisant le nombre de mutations nécessaires pour passer de l'une à l'autre. On appelle ce processus l'alignement de séquences [7]. Il existe trois types de mutations : la substitution : un nucléotide est remplacé par un autre, l'insertion : un nucléotide est ajouté dans la séquence, ou à l'inverse, la délétion : un nucléotide disparait. Le résultat de l'alignement est un score de similarité prenant en compte le nombre de nucléotides en commun et les différences entre les deux séquences alignées. L'alignement de séquences est une opération fondamentale en bioinformatique. D'ailleurs, l'outil d'alignement Blast [8] est un des outils les plus cités au monde.

À la fin des années 90, les technologies de séquençage permettent le séquençage de génomes complets, tel que *Escherichia coli* [9]. La génomique est née et il est alors possible d'étudier les gènes d'un organisme dans leur ensemble.

1.1.2 Séquençage haut-débit

Les technologies de séquençage ont subit une réelle révolution entre les années 2005 et 2008 avec l'apparition des séquenceurs de nouvelle génération (NGS). Quelques années avant, le séquençage du premier génome humain s'achevait. Cette opération a duré plus d'une dizaine d'années et couté environ 3 milliards de dollars [10]. Aujourd'hui, la technologie NGS Illumina produit le séquençage d'un génome humain en une journée et pour moins d'un millier de dollars. Cette révolution consiste donc en une augmentation conséquente du débit de séquençage, conjointement à une diminution drastique de son coût [11]. Cette thèse s'appuie sur ce type de données dont nous rappelons les spécificités.

La particularité des technologies NGS est qu'elles ne sont pas en mesure de fournir la séquence complète d'un génome mais seulement de courts extraits, appelés lectures, de quelques centaines de nucléotides selon la technologie de séquençage utilisée. La taille d'une lecture est environ 10 milles fois plus petite que le génome d'une bactérie et 30 millions de fois plus petite que la taille du génome humain. De plus, la position d'une lecture est sélectionnée aléatoirement au sein du génome visé. Pour espérer que l'intégralité du génome soit séquencé, un grand nombre de lectures doit donc être généré. Ainsi, ces lectures couvrent le génome original à une certaine profondeur. Par exemple, une couverture de 10× signifie que chaque position du génome est contenue dans 10 lectures en moyenne. Les séquenceurs NGS génèrent également un certain nombre d'erreurs de séquençage. Ces erreurs peuvent être dues à un mauvais choix de nucléotides (substitution) ou à l'oubli/ajout de nucléotides dans les lectures (délétion et insertion). La technologie Illumina domine actuellement largement le marché. Ainsi, les données sont de courtes lectures de 100 à 250 nucléotides possédant majoritairement un taux d'erreur de type substitution de l'ordre de 0.1 à 1%.

Pour récapituler, un jeu de données de séquençage se caractérise par un très grand nombre de courtes séquences d'ADN. Plus le génome visé est grand, plus il faut générer de lectures pour le couvrir en intégralité. La taille des jeux de données peut être très grande. Un jeu de données de la bactérie *E.coli* (taille du génome ≈ 5 Mbp), couvert à 116× à une taille de 1,4 Go et contient 5,3 millions de lectures. Un jeu de données humain (taille du génome ≈ 3,2 Gbp) couvert simplement à 14× atteint une taille de 115,6 Go pour 447 millions de lectures.

La nature des données NGS et la baisse du prix du séquençage sont à l'origine du phénomène de Big Data que connait la bioinformatique depuis une dizaine d'années. L'évolution des ressources informatiques gère difficilement l'augmentation exponentielle de la quantité de données génomiques (`www.genome.gov/sequencingcosts/`). Ainsi, le traitement des données NGS est devenu un secteur de recherche très actif de la bioinformatique avec pour objectifs de développer de nouvelles structures de données et de nouveaux algorithmes dédiés à leur traitement.

1.1.3 Assemblage de données génomiques

Une des opérations fondamentales en bioinformatique des séquences est l'assemblage génomique. Celui-ci a pour objectif de retrouver le génome original qui a été morcelé pendant la phase de séquençage. Un jeu de données génomiques est alors vu comme un énorme puzzle où les pièces sont les lectures. Deux lectures peuvent s'assembler si elles se chevauchent d'un certain nombre de nucléotides. Les lectures ainsi fusionnées forment des séquences de plus en plus longues que l'on appelle "contigs".

D'un point de vue informatique, ce puzzle est généralement représenté par un graphe où les nœuds sont des séquences et où les arêtes représentent le chevauchement entre deux séquences. Les contigs sont ensuite formés en parcourant les chemins de ce graphe. En pratique, cette tâche est très complexe. Inévitablement, il existe des régions du graphe où les séquences successives se

chevauchent avec plusieurs autres, forment des cycles, etc. La difficulté principale provient des grandes régions répétées du génome. Cela implique qu'une lecture peut apparaitre à plusieurs endroits du génome et possède donc plusieurs contextes.

Plus le génome original contient de longues répétitions, plus il est complexe à assembler. Plus il est long, plus le nombre de lectures nécessaire est grand et plus il faut de ressources informatiques et de temps pour l'assembler. Il est donc plus simple d'assembler les lectures d'une bactérie, dont le génome est généralement court et avec peu de longues répétitions, que celles d'un humain.

Il existe un large éventail d'outils d'assemblage. Ceux-ci diffèrent majoritairement par leur manière de représenter le graphe et de construire des contigs. Chaque méthode offre un compromis entre temps de calcul, quantité de mémoire requise et qualité des résultats. Par exemple, Celera [10] et SGA [12] utilisent un graphe où les nœuds sont les lectures. Mais ils requièrent des temps de calcul très longs pour détecter les chevauchements entre les lectures. Velvet [13] et Abyss [14] utilisent le graphe de *de-Bruijn* pour éviter cette étape. Ils passent à l'échelle sur de grands génomes. Ce graphe requiert cependant l'usage de mots de taille fixe plus petits que les lectures, appelés k-mers, ce qui a un fort impact sur les résultats de l'assemblage. Les premiers assembleurs, tels que Abyss, requièrent plusieurs centaines de Go de mémoire vive pour assembler un génome humain couvert à $30\times$ [15]. La nouvelle version de cet assembleur requiert 34 Go de mémoire et se rapproche des capacités d'une machine standard. Le temps de calcul pour effectuer l'assemblage de ce génome est de l'ordre de la journée sur une machine possédant 64 cœurs de calculs [15].

Enfin, pour rapidement se rendre compte de la difficulté du problème, on peut voir dans les statistiques de la base de données de génomes GOLD (*Genomes Online Database*) que le nombre de génomes inachevés (*draft genome*) explose par rapport à celui de génomes complets (`https://gold.jgi.doe.gov/statistics`). L'assemblage est donc encore aujourd'hui un problème ouvert et actif. Au moment de l'écriture de cette thèse, l'arrivée de la troisième génération de séquençage et ses longues lectures (pouvant atteindre des dizaines de milliers de nucléotides) lance un nouvel élan et devrait aider à résoudre le problème des répétitions [16].

1.1.4 Conclusion

La production massive de données génomiques par les technologies NGS et les avancées de la bioinformatique concernant leur traitement ont permis de déterminer de nombreux gènes et génomes. Le séquençage du génome d'un individu d'une seule espèce n'est malheureusement pas applicable à tous les organismes. La principale raison est que les séquenceurs nécessitent une quantité d'ADN importante pour produire des jeux de données qui couvrent correctement le génome. Le séquençage d'un individu eucaryote est ainsi aisé puisque ses nombreuses cellules permettent une extraction abondante de son ADN. En revanche, les espèces procaryotes nécessitent l'isolement d'un individu puis sa culture en laboratoire pour dupliquer son unique cellule. Depuis les années 1980, il est connu que certains organismes ne peuvent pas survivre en milieu contrôlé [17]. Il s'avère même qu'il s'agit d'une grande majorité des micro-organismes. Il a été estimé que moins de 0.1% des bactéries présentes dans l'eau de mer peuvent survivre à l'isolement [18]. Cela peut s'expliquer par le fait qu'il n'est pas possible de reproduire l'environnement nécessaire à la survie de ces organismes ou parce que des liaisons symbiotiques entre différentes espèces ont été rompues [19].

Le séquençage de génome à cellule unique (*Single Cell Sequencing*) est un moyen d'accéder au génome de ces organismes. Cette technique implique l'isolation d'une cellule unique puis la réalisation d'une amplification du génome avant de procéder au séquençage. Cependant, l'étape d'amplification favorise certaines régions des génomes. Seulement 40% du génome est couvert

en moyenne [20]. De plus, les techniques d'amplification peuvent introduire des mutations ou encore des séquences chimériques [21].

Une autre solution pour étudier ces organismes consiste à extraire et à séquencer tout l'ADN d'un environnement naturel [22]. En plus d'accéder aux génomes des organismes présents, cette technique permet d'étudier une communauté dans son ensemble, plutôt qu'individu par individu. Ainsi est née la métagénomique.

1.2 Métagénomique

La métagénomique est une porte d'accès au monde microbien non cultivable en laboratoire en séquencant l'ensemble de l'ADN présent dans un même milieu [23]. Le terme métagénomique a été introduit pour la première fois par Handelsman en 1998 [22] et a été rapidement adopté par la communauté scientifique. Comme dans le cas de la génomique, les NGS tirent aléatoirement un grand nombre de lectures dans l'échantillon d'ADN. La différence majeure est que cet échantillon est une "soupe" composée de nombreux individus appartenant à différentes espèces pour la plupart inconnues des bases de données biologiques. Il n'est pas possible de savoir *a priori* quelles espèces ont été séquencées. Cela complexifie d'autant plus les traitements. Par exemple, en génomique, si l'assemblage est vu comme la résolution d'un puzzle contenant des millions de pièces, l'assemblage d'un jeu de données métagénomiques peut alors être vu comme la résolution d'une multitude de puzzles mélangés. Le passage à la métagénomique soulève donc de nouveaux problèmes tels que l'identification des espèces présentes, leurs quantités, leurs fonctions dans l'environnement, et requiert le développement de méthodes spécifiques.

Dans la suite de cette section, les différents types d'analyse pour extraire des connaissances des données métagénomiques sont détaillés. Les diverses méthodes bioinformatiques sont ensuite présentées succinctement, sans prétendre être un listing exhaustif des outils existants. Avant de conclure, quelques applications métagénomiques illustrent ces analyses et les conclusions biologiques qui peuvent être tirées.

1.2.1 Questions et données

Il existe trois grands types d'analyse de jeux de données métagénomiques : taxonomique, fonctionnelle et comparative. Les **analyses taxonomiques** consistent à faire l'inventaire des taxons présents dans un environnement et à déterminer leur abondance. On cherche ici à répondre à la question : qui est présent dans mon échantillon et en quelle quantité ? De la même manière, les **analyses fonctionnelles** cherchent à déterminer les protéines et fonctions de l'environnement et leur quantité afin de répondre à la question : que peut faire la communauté ? En pratique, l'identification de la diversité taxonomique et fonctionnelle a des applications très importantes, comme la recherche d'associations entre des espèces ou des gènes avec des maladies [24]. La découverte de taxons déjà annotés fonctionnellement donne des indications sur la fonction biologique de l'environnement. Par exemple, la présence de *Cyanobacteria* suggère que la communauté est photo-synthétique [25]. La **métagénomique comparative** consiste à comparer plusieurs jeux de données afin d'en établir leur similarité globale d'un point de vue génomique. Traditionnellement, la comparaison se base sur leur composition taxonomique et/ou fonctionnelle. Plus les environnements partagent de taxons ou de fonctions, plus grande est leur similarité et *vice versa*. La comparaison peut également se baser directement sur les séquences d'ADN des environnements, c'est-à-dire sans passer par une étape d'identification de la diversité. La métagénomique comparative permet, par exemple, de clusteriser les échantillons ou d'établir des liens entre des changements de la diversité et des changements de conditions environnementales/expérimentales (individus sains, individus malades, température, etc.). La métagénomique

comparative cherche à répondre à la question : quels sont les points communs ou les différences entre les communautés ?

Il existe deux types de données métagénomiques sur lesquelles effectuer ces analyses : ciblée et plein-génome. La **métagénomique ciblée** ne s'intéresse qu'à certaines portions des génomes. À l'inverse, **la métagénomique plein-génome** (dite *shotgun* ou *Whole Genome Sequencing* dans la littérature) s'intéresse à l'ensemble du contenu génomique présent dans un échantillon d'ADN. En métagénomique ciblée, seules de très petites portions des génomes, de quelques centaines de bases, sont séquencées. Ces portions sont appelées des marqueurs phylogénétiques. Ces marqueurs ont été sélectionnés pour contenir des régions très conservées et des régions très variables. Les régions très conservées permettent de designer des amorces PCR universelles au sein des génomes afin de le séquencer. Les régions variables permettent idéalement d'identifier sans ambiguïté chaque taxon [26]. Le plus connu de ces marqueurs est l'ARNr 16S pour identifier les bactéries. Mais il existe d'autres types de marqueurs pour d'autres types de micro-organismes. Les séquences issues d'un séquençage ciblé sont appelées des amplicons car elles sont le produit d'une amplification artificielle, par réaction en chaîne par polymérase (PCR) par exemple.

Les données ciblées sont donc conçues pour estimer la diversité taxonomique d'un environnement. Pour analyser son contenu fonctionnel, il faut avoir recours aux données plein-génome. Bien sûr, les données plein-génome permettent également d'estimer la diversité taxonomique. Cependant, la métagénomique ciblée est l'approche la plus répandue pour effectuer cette analyse. La nature ciblée des données fait qu'elles sont moins complexes à traiter que des données plein-génome puisque chaque amplicon est, par définition, spécifique d'un taxon. Par conséquent, il faut un nombre beaucoup plus faible de séquences pour représenter la communauté étudiée, diminuant ainsi le coût du séquençage et le temps de traitement. À titre de comparaison, les jeux d'amplicons du projet Tara Oceans, un des plus gros projets métagénomiques actuels, contiennent quelques centaines de milliers de séquences par jeu contre quelques centaines de millions par jeu de données plein-génome. Un jeu plein-génome ne représente souvent qu'un extrait de la communauté présente. Pour être exhaustif, produire un jeu de données représentatif de la communauté d'un gramme de sol terrestre coûterait des centaines de millions de dollars [27]. En revanche, les avantages de la métagénomique ciblée sont au prix de plusieurs biais concernant l'estimation de la diversité taxonomique dont nous allons discuter dans la suite de ce chapitre.

Cette thèse s'inscrit dans la métagénomique comparative de jeux de données plein-génome. Par conséquent, une place plus importante est accordée à ce type de données et d'analyses dans l'ensemble de ce manuscrit. Ce choix d'analyse est motivé dans les sous-sections suivantes.

1.2.2 Analyse de la diversité taxonomique

Estimer la diversité taxonomique d'un jeu de données métagénomique consiste à déterminer la liste des taxons présents et à calculer leur abondance. L'**assignation taxonomique** est l'approche traditionnellement employée. Elle consiste à comparer les séquences métagénomiques à des séquences représentatives de chaque taxon, comme leur génome de référence. Par exemple, si une lecture s'aligne bien sur une région du génome de la bactérie *E. coli*, alors cette fraction du génome de *E. coli* est dite présente dans la communauté. L'inconvénient de cette approche **avec références** est qu'elle est limitée par nos connaissances.

Il existent trois techniques pour estimer la diversité taxonomique sans connaissance *a priori*. La **détection d'OTUs** (*Operational Taxonomic Units*) consiste à regrouper les amplicons via une étape de clustering. Il est alors idéalement espéré que chaque groupe corresponde à un taxon. Cette technique est dédiée à la métagénomique ciblée. Le ***binning*** est l'équivalent de la détection d'OTU pour des données plein-génome. Enfin, l'**assemblage métagénomique** est

par définition la méthode ultime pour estimer la diversité d'un échantillon. Cependant, à ce jour, il n'existe pas d'outil générique capable de déterminer l'ensemble des génomes complets présents dans ces données. Mais nous allons voir que l'assemblage métagénomique et le *binning* sont intimement liés. Si ces trois approches ne répondent pas explicitement à la question "qui est présent dans ma communauté ?", celles-ci peuvent donner de bonnes indications de diversité, comme le nombre et le type de taxons distincts présents dans la communauté. L'assignation taxonomique peut également être basée sur le résultat de ces méthodes.

Une fois la liste des taxons identifiée, leur abondance est estimée en fonction du nombre de séquences alignées sur leur référence. Dans le cas d'une estimation de la diversité sans références, l'abondance des OTUs (ou *bins*) est estimée en fonction du nombre de séquences qui constituent leur cluster.

1.2.2.1 Assignation taxonomique

Une méthode d'assignation taxonomique cherche à attribuer chaque séquence métagénomique à un taxon. L'approche classique consiste à aligner toutes les séquences contre toutes celles d'une ou plusieurs banques de références. Le score d'alignement le plus élevé entre une séquence et une référence est alors utilisé pour estimer leur ressemblance d'un point de vue phylogénétique. Il existe de nombreuses banques de références, telles que RefSeq [28] pour aligner les séquences contre des génomes complets, ou RDP [29] et Greengenes [30] pour les aligner contre des marqueurs phylogénétiques connus.

Dans le cas de données ciblées, cette approche est exploitable puisque les jeux d'amplicons ont une taille modérée. De plus, l'alignement est rapide car les références sont également de courts marqueurs. Cependant, les outils d'assignation taxonomique d'amplicons, tels que Qiime [31] et Mothur [32], fonctionnent autrement : ils détectent tout d'abord les OTUs (section 1.2.2.2), puis élisent un amplicon représentatif par OTU qui est alors aligné contre les références.

Les limitations de cette technique d'assignation, en matière de passage à l'échelle, apparaissent lorsque des dizaines de millions de lectures d'un simple jeu plein-génome doivent être alignées contre des génomes complets (plus d'une dizaine de milliers de génomes de référence actuellement). Par exemple, le logiciel Megan [33] base son assignation sur la sortie de Blast. L'usage d'outils d'alignement spécialisés pour les données NGS, tels que BWA [34] et Bowtie2 [35], améliorent drastiquement les temps de traitement mais ils ne passent pas à l'échelle au regard des calculs demandés.

Pour pallier ce problème, d'autres outils, dits sans-alignement (*alignment-free*), ont récemment émergé. La comparaison de séquences sans-alignement est un domaine de recherche à part entière datant des années 80 [36] mais dont la popularité a particulièrement augmenté suite à l'apparition des technologies NGS. Cette approche fonctionne généralement en utilisant le k-mer (mot de taille k) comme unité de comparaison. Une mesure de similarité simple entre deux séquences peut être leur pourcentage de k-mers en commun. L'avantage des k-mers est qu'ils s'indexent et se comparent extrêmement rapidement. Kraken [37] est un des premiers outils à avoir implémenté cette idée et conduit à une assignation extrêmement rapide des lectures. En plus d'un catalogue de génomes de référence, Kraken utilise leur classification taxonomique pour assigner les lectures. Pour une lecture donnée, Kraken regarde dans quels génomes ses k-mers apparaissent. La lecture est alors assignée à l'ancêtre commun le plus proche de ces génomes d'après l'arbre taxonomique. Au moment de sa publication, la vitesse de Kraken (4.1 millions de lectures par minute, 909 fois plus rapide que MegaBlast [38], une version rapide de Blast) et la qualité de son assignation en ont fait un des standards. Cependant, celui-ci à une empreinte mémoire élevée (70 Go) pour stocker un index contenant seulement 2787 génomes bactériens. Depuis la sortie de Kraken, l'assignation sans-alignement a été un domaine de recherche très

actif. De nombreux outils ont été publiés, comme Centrifuge [39] ou Kaiju [40].

Une autre catégorie d'outils estime la diversité des jeux plein-génome en n'utilisant qu'une fraction des lectures : celles qui codent pour un marqueur phylogénétique donné, tel que l'ARNr 16S. Cela revient en quelque sorte à réduire les jeux plein-génome à des jeux de métagénomique ciblée. Par exemple, dans le pipeline d'analyses métagénomiques EMG [41], le logiciel Hmmer [42] est tout d'abord utilisé pour détecter les lectures codant pour des ARNr. Celles-ci sont ensuite traitées par des logiciels classiques d'analyse d'amplicons tel que Qiime. Dans la même idée, Metaphlan [43, 44] réduit les banques de références à environ un million de courts marqueurs très conservés et spécifiques des clades. L'alignement des lectures contre ces marqueurs est ensuite effectué grâce à Bowtie2. Kraken reste plus rapide que Metahplan (11 fois plus rapide [37]) grâce à l'utilisation des k-mers. Cependant, l'index de Metaphlan (1 Go), correspondant à environ 17 000 génomes de référence, lui permet d'être utilisé sur des machines standards.

Conclusion. L'assignation taxonomique est un domaine extrêmement actif de la métagénomique. Il existe aujourd'hui un large éventail d'outils qui offrent de bons compromis entre temps de traitement, ressources informatiques requises et qualité d'assignation.

Cependant, les approches avec références n'annotent qu'une fraction des séquences métagénomiques puisqu'une grande majorité des espèces nous est encore inconnue (plus de 99% d'entre elles [45]). Par exemple, dans le cadre de l'analyse du microbiome humain, le consortium HMP a aligné plus de 38 milliards de lectures de haute qualité sur plus de 3000 génomes de référence complets à 80% de similarité minimum. Seulement 57% d'entre elles ont atteint une cible [46], alors que c'est pourtant un environnement très étudié (voir `https://www.hmpdacc.org/hmp/publications.php` par exemple). De plus, pour les séquences provenant de taxons inconnus, le risque est de les assigner à de mauvais taxons qui leur seraient proches d'un point de vue génomique. Enfin, ce nombre de séquences non assignées peut drastiquement augmenter dans des environnements comme l'eau de mer ou le sol, qui sont non seulement plus complexes mais aussi bien moins étudiés. Cette importante fraction de séquences ignorées peut fortement biaiser le résultat des analyses et les conclusions qui en sont tirées. Pour pallier ce biais, d'autres approches sont mises en œuvre : la détection d'OTUs, l'assemblage métagénomique et le *binning*.

1.2.2.2 Détection d'OTUs

La détection d'OTUs consiste à clusteriser les amplicons. Idéalement, il est espéré que chaque OTU corresponde à un taxon. Cette technique est dédiée aux données de métagénomique ciblée. Puisque les séquences ciblées proviennent de la même région de chaque génome, sur quelques centaines de nucléotides, celles-ci sont directement comparées toutes contre toutes sur la base de leur similarité. Un seuil d'identité est ensuite appliqué pour les regrouper en OTUs. Intuitivement, un seuil d'identité élevé génère des OTUs correspondant à un rang taxonomique plus précis et inversement. Un seuil d'identité à 97% est souvent utilisé pour obtenir des OTUs au niveau de l'espèce [47]. Les techniques existantes s'attachent donc à optimiser ces comparaisons de séquences toutes contre toutes.

Cette approche a été optimisée par le logiciel Esprit [48] en évitant l'alignement des paires d'amplicons pour lesquels il est facile de déduire que la similarité sera en dessous du seuil d'identité minimum à partir de leur composition en k-mers. En effet, si le seuil d'identité est fourni *a priori*, il est possible de déterminer le nombre minimum de k-mers que les séquences doivent partager pour avoir une identité au moins égale au seuil. Cette approche est particulièrement efficace si le seuil d'identité est élevé, ce qui est souvent le cas. De nombreux outils de clustering

et d'alignement de séquences implémentent cette idée.

Les approches gloutonnes clusterisent les amplicons en une seule passe sur les données. Au départ, il n'y a pas encore de cluster donc le premier amplicon devient automatiquement le centre d'un premier cluster. Le deuxième amplicon est regroupé avec le premier si les deux amplicons sont similaires. Dans le cas contraire, il devient le centre d'un nouveau cluster. Chaque amplicon qui suit est comparé aux centres et s'associe avec un cluster existant ou en crée un nouveau. Des outils très populaires de la bioinformatique existent pour effectuer un tel clustering, tels que CD-HIT [49] et UCLUST [50]. Cependant, ces deux outils n'ont pas spécifiquement été optimisés pour le clustering d'amplicons.

SUMACLUST [51] et SWARM [52] sont deux outils dédiés à la détection d'OTUs. Ceux-ci proposent de trier les amplicons par abondance en pré-traitement (l'abondance d'un amplicon est égale à son nombre de copies exactes). En effet, les amplicons abondants ont moins de chance d'être erronés (erreurs dues à la technologie de séquençage ou à l'amplification) et devraient finir au centre de leur cluster. L'outil SWARM s'abstrait du seuil d'identité global arbitraire. Pour cela, un grand seuil d'identité est tout d'abord choisi et diminue itérativement tant qu'il reste des amplicons à affecter. Les clusters croissent ainsi jusqu'à leur limite naturelle et sont fermés dès qu'ils n'agglomèrent plus d'amplicons. SUMACLUST et SWARM raffinent également leurs clusters en exploitant l'abondance des amplicons et la topologie d'un graphe où les noeuds sont les séquences et les arêtes la similarité. Les longs enchainements d'un amplicon dont l'abondance décroit, puis croit (forme de vallée), sont scindés pour créer des clusters de meilleure résolution. Ces deux outils utilisent également les instructions SIMD (Single Instruction Multiple Data) des processeurs modernes pour comparer les amplicons en parallèle à grain fin. Sur un jeu de données de plus de deux milliards d'amplicons, SWARM détecte les OTUs en 3h41. Il est beaucoup plus rapide que UCLUST dont le temps de calcul est estimé à 150 jours.

Conclusion. Les avantages de la métagénomique ciblée, en termes de coût de séquençage, de simplicité d'analyse et de temps de calcul, sont au prix du biais généré par l'utilisation d'un seul marqueur pour estimer la diversité. De nombreuses études évoquent les inconvénients de cette approche : le marqueur peut ne pas être assez informatif pour discriminer les sous-espèces ou souches [53, 54, 55] ; plusieurs espèces peuvent être représentées par la même séquence de marqueur phylogénétique [56] ; l'étape d'amplification favorise certaines régions génomiques et biaise donc l'abondance des amplicons [57] ; les erreurs d'amplification génèrent des amplicons erronés difficilement détectables ; la possible présence de multiples copies d'un même marqueur dans un même génome biaise son abondance [58] ; enfin, les résultats, en termes d'OTUs identifiés, ne sont pas homogènes en fonction du choix des marqueurs [59, 60].

La détection d'OTUs peut être appliquée aux données plein-génome via un pré-traitement. Pour cela, un logiciel de détection des marqueurs phylogénétiques, tel que HMMER [42], est utilisé afin de réduire un jeu plein-génome à un jeu de données ciblées. Cependant, il existe d'autres approches pour estimer la diversité taxonomique sans référence spécifiques aux données plein-génome : l'assemblage métagénomique et le *binning*.

1.2.2.3 Assemblage métagénomique

Comme dans le cas de la génomique, l'assemblage métagénomique a pour but de reconstruire les génomes des espèces présentes qui ont été morcelés pendant le processus de séquençage. L'assemblage génomique des données d'un unique individu d'une seule espèce est déjà un problème très complexe (section 1.1.3). Il s'agit maintenant d'assembler les séquences d'ADN de nombreux individus appartenant à diverses espèces. Les assembleurs génomiques requièrent généralement une couverture minimum ($> 20\times$) et uniforme afin de générer une référence complète.

En métagénomique, la couverture des espèces, représentée par leur abondance dans le milieu, est hétérogène et souvent très faible. Parmi les espèces présentes, certaines peuvent avoir des génomes proches et risquent d'être fusionnées pendant l'assemblage, générant ainsi des contigs chimériques [61]. Une autre source de difficulté provient des variations engendrées par le polymorphisme des multiples individus d'une même espèce. Il est très complexe d'établir un génome consensus à partir des génomes de différents individus d'une même espèce [62].

Si la reconstitution des génomes complets parait impossible dans de telles circonstances, l'assemblage métagénomique reste très utile pour obtenir des séquences plus longues que les lectures. Les contigs améliorent généralement les analyses métagénomiques telles que le *binning*, l'assignation taxonomique [63] ou la détection de gènes [24]. Pour cette tâche, des assembleurs génomiques classiques peuvent être lancés directement sur les données métagénomiques. Cependant, il existe aujourd'hui quelques assembleurs qui s'attaquent à certains problèmes spécifiques de la métagénomique et qui obtiennent généralement des contigs plus longs.

Une idée notable pour améliorer les assembleurs génomiques classiques est de prendre en compte l'abondance des espèces, en plus de leur séquence, pour différencier ces espèces. Cette idée a été introduite par l'assembleur MetaVelvet [64] qui est une extension de l'assembleur Velvet pour des données métagénomiques. L'abondance des espèces peut être estimée en comptant les k-mers du jeu de données en utilisant un grand k ($k > 21$) [65, 66]. Lorsque l'assembleur a plusieurs choix pour prolonger un contig (il se situe dans une région génomique présente chez plusieurs espèces par exemple), il choisit le chemin qui a l'abondance la plus proche de celle du contig courant.

Conclusion. L'assemblage des génomes complets basé sur de courtes lectures métagénomiques reste aujourd'hui un problème ouvert, voire inaccessible dans le cas d'environnements complexes, tels que le sol ou l'eau de mer, constitués de nombreuses espèces rares et d'un taux de polymorphisme élevé [67].

La compétition de traitement de données métagénomiques CAMI [68] a récemment mis en lumière la difficulté de ce problème sur un jeu de données simulées contenant environ 600 génomes ($\sim$500 millions de lectures de 150 bp). Six assembleurs ont été évalués et ont rendu des résultats hétérogènes. Les meilleurs d'entre eux ont généré environ 16 fois plus de contigs qu'attendu et environ 30% des génomes n'ont pas été assemblés. En particulier, la distinction des souches (génomes $> 95\%$ de similarité) n'a pas du tout été résolu. Seulement 22% de ces génomes ont correctement été assemblés au mieux. De manière attendue, aucun assembleur n'a réussi à assembler les espèces rares ($< 5\times$ de couverture).

L'assemblage métagénomique reste utile dans le but de produire des séquences plus longues que les lectures. Cependant, certains problèmes n'ayant pas été résolus, les contigs ne représenteront qu'une partie des données originales.

1.2.2.4 Binning

Le clustering de lectures métagénomiques sans référence, appelé *binning* dans ce manuscrit, consiste à attribuer un groupe, appelé *bin*, à chaque lecture. C'est l'équivalent du clustering des amplicons en OTU mais pour des données plein-génome. Il est donc espéré que chaque groupe corresponde à un taxon. Le *binning* est cependant beaucoup plus complexe que le clustering d'amplicons. Premièrement, d'un point de vue informatique : un échantillon plein-génome peut contenir des dizaines, voire des centaines de millions de lectures, contre les centaines de milliers d'amplicons des jeux de données ciblées. Deuxièmement, les données sont beaucoup plus hétérogènes car elles ne proviennent pas simplement d'une région ciblée du génome. Il ne s'agit donc pas essentiellement d'avoir recours à de l'alignement de séquences car deux lectures peuvent être

parfaitement dissimilaires mais appartenir au même génome.

La plupart des outils de *binning* comparent les lectures métagénomiques en se basant sur leur composition en petits k-mers ($k \approx 4$). Cette composition est représentée par un vecteur de k-mers, de taille 4^k, chacun associé avec son nombre d'occurrences dans la séquence. Ces vecteurs peuvent ensuite être comparés, en utilisant la distance de Spearman par exemple, puis clusterisés avec des techniques d'apprentissage non-supervisé. Cette approche se base sur l'observation que la composition en k-mers est stable au sein du génome d'une espèce mais varie entre les génomes d'espèces distinctes [69]. Celle-ci a été implémentée pour la première fois dans l'outil TETRA [70]. Cependant, cette méthode ne fonctionne efficacement que si la taille des séquences est au moins de 1000 nucléotides [71].

Pour répondre à ce problème, MetaCluster-4 [72] regroupe, dans un premier temps, les lectures qui partagent un grand k-mer ($k > 35$). Plus k est grand, plus la probabilité que deux lectures partageant un k-mer proviennent du même génome augmente. Il existe tout de même une limite à partir de laquelle les k-mers deviennent trop sensibles aux variants génomique et aux erreurs de séquençage. Ces groupes de lectures sont ensuite eux-mêmes regroupés en extrayant leur représentation en 4-mers, puis en les clusterisant avec l'algorithme *k-means* [73]. Basé sur l'observation que le *binning* est sensible à l'abondance des espèces [66], notamment les espèces très abondantes tendent à créer de nombreux *bins*, MetaCluster-5 [72] sépare dans un premier temps les lectures provenant d'espèces abondantes des espèces rares via leur abondance en grand k-mers ($k \approx 30$). L'abondance des grand k-mers est, en effet, un bon estimateur de l'abondance des espèces [66]. Un algorithme *k-means* est ensuite effectué pour regrouper les séquences en utilisant leur représentation en 5-mers pour les lectures abondantes et en 4-mers pour les lectures rares. Dans sa dernière version MetaCluster-TA s'appuie en plus sur l'information de contigs pour regrouper les lectures. Plus récemment, de nouvelles approches, telles que CONCOCT [74] et MetaBAT [75], ont amélioré le *binning* en intégrant l'information de l'abondance de contigs à travers de multiple jeux de données (co-abondance) en plus de l'information de leur composition.

Conclusion. Un des principaux goulots d'étranglement actuel des approches de *binning* vient de son lien intime avec l'assemblage métagénomique. Cette étape est requise pour étendre les courtes lectures et comparer efficacement leur composition. Dès lors, une large fraction des lectures peut être négligée comme nous l'avons vu précédemment. L'assemblage est également coûteux en temps et en ressources informatiques.

Comme pour l'assemblage métagénomique, la compétition CAMI [68] a évalué la qualité de 5 outils de *binning*. L'évaluation a été basée sur la capacité de ces outils à regrouper correctement des contigs générés à partir de différents jeux de données simulés à différents niveaux de complexité : faible (40 génomes), moyen (132 génomes) et élevé (596 génomes). Pour chaque *bin* attendu, des mesures de rappel et de précision ont été calculées représentant respectivement la capacité des outils à retrouver les contigs appartenant à ce *bin* et à ne pas attirer les contigs des autres *bin*. Comme pour les assembleurs, les outils de *binning* ont eu des résultats très hétérogènes. Les meilleurs résultats ont été obtenus par les outils récents s'appuyant sur la co-abondance des contigs. Sur les jeux à complexité faible et moyenne, le meilleur outil a eu 75% de rappel et 92% de précision. Contrairement aux outils d'assemblage, les résultat globaux se sont fortement dégradés sur les jeux à complexité élevée avec un rappel descendant à 50%. Les différents outils invitent à un compromis entre la quantité de données considérée et la qualité du *binning*. Les outils ayant la meilleure précision n'ont considéré que 50% des données, alors que ceux qui ont considéré le plus de données ($> 90\%$) ont mal assigné 20 à 50% des séquences.

1.2.3 Analyse de la diversité fonctionnelle

Déterminer la diversité fonctionnelle d'un métagénome consiste à faire le listing des fonctions exercées par celui-ci associées à leur abondance comme dans le cas de l'estimation de la diversité taxonomique. Cette analyse n'est applicable qu'à des jeux de données plein-génome. Elle peut être découpée en deux étapes : la prédiction de gènes et l'annotation fonctionnelle. La prédiction de gènes consiste à détecter les lectures qui codent pour une protéine. Une fois ces lectures détectées, la fonction du gène est déterminée en fouillant les bases de données protéiques. Ces deux étapes ne sont pas exclusives et peuvent être effectuées en même temps.

Prédiction de gènes. Comme pour l'estimation de la diversité taxonomique, l'approche simple pour déterminer les lectures codantes consiste à les aligner contre un ensemble de gènes connus. Si une lecture est similaire à un gène, elle est dite codante et on peut simultanément lui attribuer les fonctions du gène. De manière similaire, les lectures peuvent être traduites dans l'alphabet protéique (6 traductions possibles par lecture) et recherchées dans les bases de données protéiques directement. Des outils alignent les séquences tout en effectuant cette traduction à la volée, tel que BLASTX [76].

Il existe également des méthodes qui détectent de nouveaux gènes, comme ORPHELIA [77, 78]. Pour cela des algorithmes d'apprentissage sont entrainés à reconnaitre des patterns de gènes connus. Ils prennent en comptent la taille, les codons à partir desquels commence et se termine la traduction en protéines (codons start et stop), les biais de GC, etc.

Annotation fonctionnelle. Une fois les lectures codantes prédites, leurs fonctions peuvent être recherchées en requêtant les bases de données protéiques.

Certaines bases de données sont structurées autour du concept de famille protéique. Une famille protéique réunit les protéines issues de gènes ayant divergés d'un ancêtre commun et qui sont donc susceptibles d'exercer les mêmes fonctions. La comparaison d'une lecture à une famille requiert soit de la comparer à toutes les séquences protéiques constituant la famille, soit à un modèle statistique représentant la diversité de la famille (un modèle HMM [79] par exemple). Si une lecture est similaire à une famille alors on lui attribue ses fonctions. Les protéines peuvent également être découpées en domaines structuraux et fonctionnels, avec des méthodes comme PFAM [80], afin d'améliorer l'annotation.

Les bases de données protéiques sont nombreuses et ne sont que partiellement redondantes. Ainsi, la base de données UNIPROT [81] est un entrepôt où les séquences protéiques peuvent être déposées par les équipes scientifiques du monde entier. UNIPROT fait le lien entre de nombreuses bases de données spécialisées. Le programme INTERPRO [82] intègre ces bases de données protéiques, les organise en familles et prédit leurs domaines. La base de données KEGG [83] est quant à elle un ensemble de bases de données regroupant différents types de données biologiques (génome, gène, protéine, réseau métabolique, etc) et délivre une annotation à différents niveaux.

Conclusion. Comme pour l'assignation taxonomique, la détermination des fonctions d'une communauté est très dépendante des banques de références. Cependant, la détection d'un gène dans un environnement métagénomique ne signifie pas qu'il était exprimé au moment du prélèvement.

Pour résoudre le problème, on peut maintenant directement séquencer le transcriptome d'une communauté de micro-organismes à un instant donné. La métatranscriptomique est le domaine de la métagénomique qui analyse les données des gènes exprimés. C'est un domaine encore récent. Le premier transcriptome environnemental ne date que de 2005 [84]. Coupler les analyses métagénomiques et métatranscriptomiques offre une meilleure exploration de la structure et la

fonction de communautés microbiennes [85]. Par exemple, les données métatranscriptomiques permettent de valider l'annotation de gènes non présents dans les banques de références mais détectés dans les données métagénomiques.

1.2.4 Métagénomique comparative

La métagénomique comparative compare plusieurs échantillons métagénomiques afin d'en dériver des mesures de similarité. Le résultat est une matrice de similarité, communément appelée matrice de distances, de taille $N \times N$ où N est le nombre de jeux de données à étudier. Chaque cellule indique la similarité (ou inversement la distance ou dissimilarité) entre une paire d'échantillons. Il existe deux types de métagénomique comparative : celle qui se base sur la diversité identifiée grâce aux techniques présentées précédemment, et la métagénomique comparative *de novo* qui compare directement les séquences des jeux de données.

Avec estimation de la diversité. La diversité extraite de chaque jeu de données (avec ou sans références) peut être vu comme un ensemble d'éléments (taxons, gènes, OTU ou *bin*) associés à leur abondance. Pour une paire de jeux donnée, la similarité dépend du nombre d'éléments communs et spécifiques. Mais la valeur de similarité finale dépend de la définition de l'indice de similarité utilisé. Ces indices sont nombreux et se regroupent en deux familles (voir [86] pour une classification plus fine) : les indices qualitatifs et les indices quantitatifs. La première famille traite les éléments de manière égale, qu'ils soient rares ou très abondants, et se base sur le nombre d'éléments spécifiques et partagés entre les paires d'échantillons. Dans cette catégorie, nous retrouvons le classique index de Jaccard. À l'inverse, les indices quantitatifs s'appuient sur les variations d'abondance des éléments. Ici, les éléments abondants ont plus de poids que les éléments rares. Si tous les éléments sont partagés entre les communautés, cette approche peut tout de même les différencier via leurs différences d'abondance. La dissimilarité de Bray-Curtis [87] est un des indices les plus populaires de cette catégorie.

La méthode Unifrac [88] sort de ce lot. Elle intègre l'information des liens de parenté entre les membres des communautés. Ces liens sont représentés par la distance phylogénétique entre les organismes. Unifrac infère un arbre phylogénétique à partir des séquences des OTUs détectés. Dans cet arbre, une branche est dite commune aux deux échantillons s'il existe un OTU de chacune des deux communautés parmi ses descendants. La mesure de similarité Unifrac est donné par la fraction de longueur de branches partagées.

Métagénomique comparative *de novo*. La métagénomique comparative *de novo* compare les paires d'échantillons sans estimation de la diversité taxonomique ou fonctionnelle. L'objectif est d'estimer le contenu génomique partagé. Pour cela, les lectures du premier jeu sont alignées contre les lectures du second. Les paires de séquences ayant une forte identité sont dites similaires et on peut émettre l'hypothèse qu'elles proviennent du même génome. La similarité entre deux échantillons est alors définie par leur pourcentage de séquences similaires. Ceci n'est qu'un exemple de mesure de similarité sans estimation de la diversité. Cette thèse s'inscrit dans la métagénomique comparative *de novo* et le chapitre 2 présente l'état de l'art.

La métagénomique comparative *de novo* doit correspondre à une comparaison des jeux à un niveau taxonomique. En effet, les lectures proviennent avant tout du génome des espèces présentes, et non spécifiquement de leurs gènes.

1.2.5 Applications

Quelques projets métagénomiques sont présentés pour expliciter les différents types d'analyses. Puisqu'un jeu de données correspond à l'image d'une communauté à un instant, espace et

conditions donnés, les projets métagénomiques sont généralement constitués de plusieurs jeux. Par exemple, si un projet métagénomique veut explorer exhaustivement le microbiome intestinal humain et tirer des conclusions quant à sa composition taxonomique, il faut considérer plusieurs individus aux caractéristiques différentes qui sont susceptibles de faire varier cette composition (sexes, âges, origines, habitudes alimentaires, maladies, etc). Un jeu métagénomique n'est ainsi pas simplement un ensemble de séquences d'ADN. Il se compose également d'un ensemble de métadonnées correspondant aux conditions de prélèvement. Ces métadonnées et les résultats des analyses métagénomiques sont combinés pour extraire des connaissances.

Exploration du microbiome humain. Le projet MetaHIT a eu pour objectif d'explorer le microbiome intestinal humain et de déchiffrer les relations entre les micro-organismes et la santé humaine [2]. Ce projet a séquencé 124 individus sains et malades (540 Gbp de séquences d'ADN). Un catalogue de 3.3 millions de gènes bactériens en a été extrait. Parmi les 19 000 fonctions détectées dans ce catalogue, 5 000 n'avaient jamais été trouvées avant. Cela illustre les apports de la métagénomique pour étudier les environnements microbiens. Le profil taxonomique des jeux de données a été défini en alignant les lectures sur un catalogue de 1511 génomes de référence. Ces profils ont révélé qu'environ 1000 espèces distinctes sont présentes dans l'intestin humain. Cependant, un individu seul abrite environ 170 espèces. La comparaison des profils taxonomiques a montré l'existence de 3 enterotypes [89], chacun caractérisé par une forte abondance de différents taxons : *Bacteroides*, *Prevotella* ou *Ruminococcaceae*. Par ailleurs, ces enterotypes seraient liés au régime alimentaire des individus plutôt qu'à leurs caractéristiques (age, poids, nationalité, maladies, etc) [90].

Le projet Human Microbiome Project (HMP) [46, 1] a étendu cette exploration à l'ensemble du microbiome humain et à une plus grande cohorte. Celui-ci a généré plus de 5000 échantillons ciblés (2.6 Go de données) et plus de 600 échantillons plein-génome (3.5 To de données) extraits à partir des cavités buccales et nasales, de l'intestin, de la peau et des organes urogénitaux de 242 adultes en bonne santé. La comparaison taxonomique de ces échantillons a permis de découvrir qu'il y a une forte variabilité d'organismes entre les individus [91] et a écarté la théorie d'un noyau universel d'organismes partagés par l'homme en analogie à la large proportion d'ADN que nous partageons. En revanche, des fonctions nécessaires à la vie microbienne sont constamment présentes dans tous les tissus et forment donc un noyau fonctionnel universel [92, 1].

Partant du principe que chaque individu possède sa propre communauté microbienne [93, 91] et qu'elle se propage (respiration, contacts, etc. [94]), le projet Home Microbiome Project [95] a recherché des liens entre les communautés bactériennes d'individus d'une même famille et de leur maison, ainsi qu'entre différentes familles. Plus de 1625 échantillons ciblés (environ 15 millions d'amplicons) ont été récoltés au sein de sept familles américaines et sur différentes surfaces de leur maison pendant 6 semaines. Leur annotation taxonomique et leurs comparaisons ont été effectuées avec le logiciel Qiime. Ces comparaisons ont montré que les communautés bactériennes diffèrent entre chaque maison et que le microbiome d'une maison est largement lié à celui de ses occupants. Pour confirmer cela, plusieurs familles se sont déplacées dans d'autres maisons. Il a été observé que les communautés bactériennes des maisons convergent rapidement vers celles de ses nouveaux occupants.

Exploration du biome terrestre. Le projet MetaSoil a eu pour objectif de caractériser la diversité d'un sol sain (Park Grass Environment of Rothamsted, England) préservé de toutes techniques modernes d'agriculture (engrais, désherbants, etc.). Treize métagénomes ont été séquencés dans différentes conditions et ont été analysés afin d'étudier l'impact sur la diversité de la profondeur sous la terre de l'échantillonnage, des changements saisonniers et de différents protocoles de séquençage. L'annotation taxonomique et fonctionnelle de ces échantillons et leurs

comparaisons, effectuées par le serveur MG-RAST [96], a révélé une importante corrélation entre la classification des échantillons et la technique de séquençage employée, indiquant d'importants biais lors de cette étape [97]. En revanche, aucune corrélation au niveau fonctionnel n'a pu être établie concernant la profondeur et le rythme des saisons [98]. Les auteurs de cette étude insistent particulièrement sur l'importance de mener des études comparatives sur plusieurs échantillons, récoltés dans différentes conditions, afin d'extraire plus d'informations qu'en se concentrant sur l'étude d'un unique jeu de données [99].

Exploration de la biodiversité océanique. Le projet Global Ocean Sampling (GOS), qui s'est attaqué à l'exploration de la biodiversité océanique dans les années 2000, a introduit la métagénomique comparative *de novo* pour la première fois. Ce projet a séquencé 44 jeux de données métagénomiques plein-génome provenant de 41 positions géographiques différentes. La métagénomique comparative *de novo* a été employée compte tenu des faibles connaissances dont nous disposons concernant le milieu océanique. La matrice de similarité résultante a été utilisée pour clusteriser les échantillons hiérarchiquement. Ce clustering montre que les échantillons partageant des facteurs environnementaux communs ont tendance à se regrouper. Par exemple, la heatmap résultante (figure 1.2) montre très clairement que les échantillons tendent à se regrouper selon leur proximité géographique [100].

1.3 Conclusion

Actuellement, les méthodes usuelles pour extraire des connaissances des jeux métagénomiques se basent sur les banques de références. Cette approche est la plus informative pour un biologiste mais peut ne pas considérer une grande partie des séquences appartenant à des espèces ou gènes encore inconnus. Même dans le cas du microbiome humain, qui est bien étudié, une large fraction des lectures peut ne pas être assignée aux génomes de références (section 1.2.2).

Lorsqu'une assignation précise n'est pas envisageable, les jeux peuvent néanmoins être comparés sur la base de leur diversité taxonomique. Cette diversité peut être estimée en séquencant seulement des marqueurs phylogénétiques, tel que l'ARNr 16S, et en les regroupant en OTUs. Cependant, cette approche souffre de nombreux biais et ne capture pas toute la diversité d'un échantillon. Par exemple, plusieurs espèces peuvent partager le même marqueur (section 1.2.2.2). Malheureusement, cette même approche sur des jeux de données plein-génome, en prenant en compte l'ensemble des lectures, est en pratique quasiment inexploitable pour au moins deux raisons : le *binning*, souvent associé à une étape d'assemblage métagénomique, est une tâche très coûteuse en temps de calcul et en ressources informatiques, et omet une large fraction des lectures. Ces limitations ont tendance à s'accentuer avec l'augmentation de la complexité des milieux étudiés (sections 1.2.2.3 et 1.2.2.4).

Dans ce contexte, il est plus pratique d'abandonner les techniques d'estimation de la diversité et de comparer les communautés en se basant directement sur les séquences brutes des jeux de données métagénomiques. Cette approche, appelée métagénomique comparative *de novo*, a pour la première fois été mise en œuvre dans le cadre du projet océanique Global Ocean Sampling (GOS) dans les années 2000. Dans le cas du projet GOS, une telle approche a été possible compte tenu de la taille modérée des jeux de données : un échantillon contenait en moyenne 175 milliers de lectures de $\sim$1250 bp. Certains projets actuels, séquencés avec la technologie Illumina, se situent à une toute autre échelle en termes de volume de données. Par exemple, le projet océanique Tara Oceans, actuellement le plus gros projet de séquençage métagénomique, représente un total de 644 échantillons métagénomiques plein-génome. Soit 240 milliards de lectures ou 24 trillions de bases. De plus, cette expédition n'en est qu'à ses débuts, des milliers de jeux de données seront prochainement disponibles pour explorer exhaustivement la biodiversité

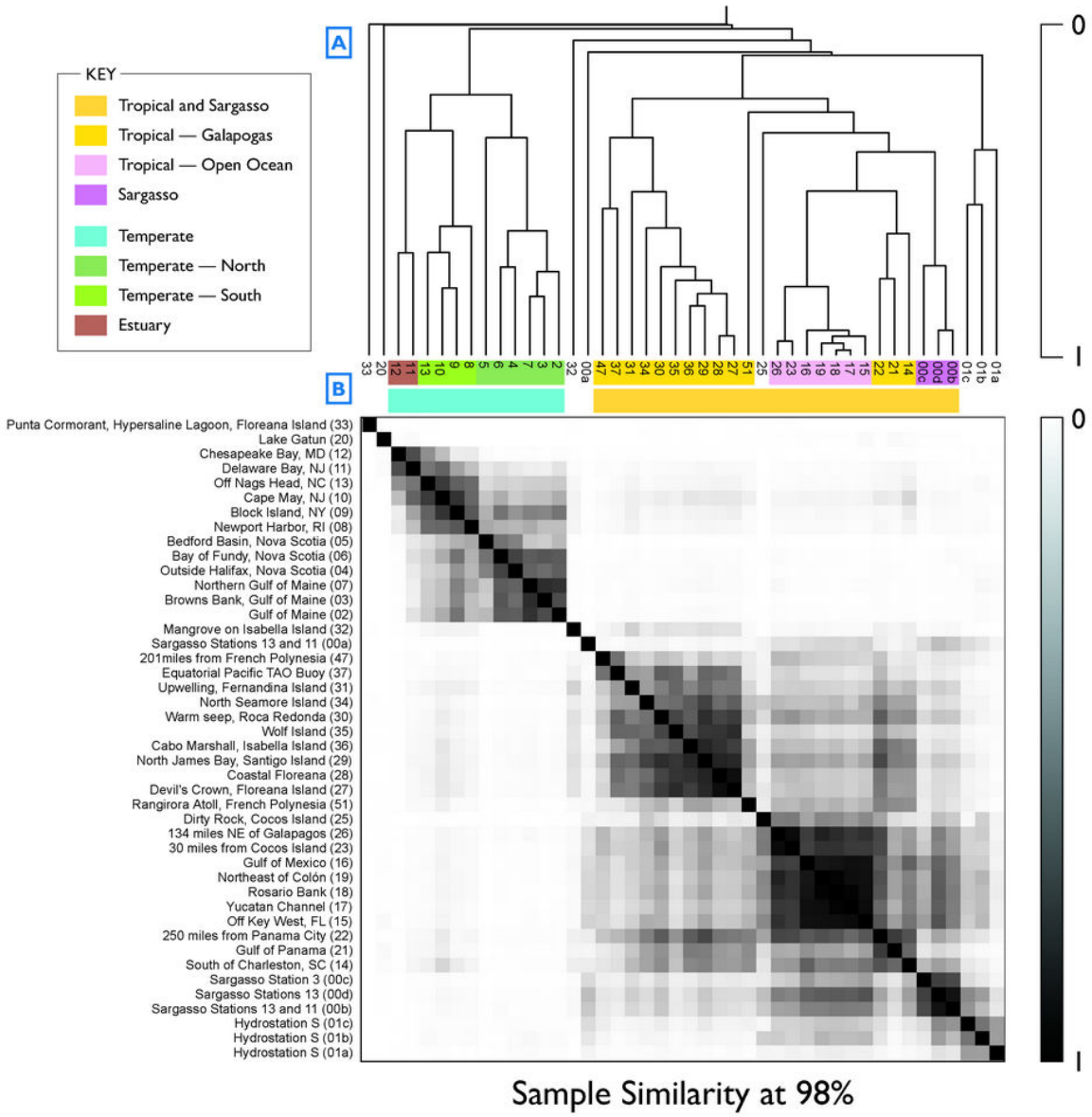

Figure 1.2. Similarité entre les échantillons en termes de contenu génomique partagé. (**A**) Clustering hiérarchique des échantillons basé sur la matrice de similarité. (**B**) Représentation de la matrice de similarité sous la forme d'une heatmap. Chaque cellule représente la similarité entre une paire d'échantillons. Plus sombre est la cellule, plus grande est la similarité. Les lignes et colonnes ont été ordonnées selon les résultats du clustering hiérarchique (figure extraite de Venter *et al.* [100]).

océanique. Appliquer l'approche d'alignement toutes séquences contre toutes utilisée dans le projet GOS est actuellement hors de portée sur une telle masse de données. De plus, Tara Oceans n'est qu'un exemple de projet métagénomique de grande envergure. Le projet de séquençage MetaSUB [101] vise à étudier le microbiome urbain dans différentes villes du monde. Des projets de séquençage collaboratif ont également vu le jour comme l'Ocean Sampling Day [102]. Ce projet a permis le prélèvement et le séquençage simultané de 191 échantillons métagénomiques à travers le monde. Similairement, le projet MetaSUB a appelé à un Global City Sampling Day pour générer des données métagénomiques urbaines.

L'objectif principal de cette thèse est de développer une nouvelle méthode de métagénomique comparative *de novo* permettant de passer à l'échelle sur la comparaison de projets métagénomiques de grande envergure.

1.4 Organisation du manuscrit

Le chapitre 2 expose l'état de l'art de la métagénomique comparative *de novo*. Les méthodes présentées comparent les jeux de données sans connaissance *a priori* et sans estimation de la diversité taxonomique ou fonctionnelle.

Le chapitre 3 présente Simka, notre nouvelle méthode pour traiter des projets métagénomiques de grande envergure. Notre proposition pour comparer les communautés est de remplacer les comptages d'espèce par des comptages de grands k-mers. Les performances de notre nouvel outil, d'un point de vue ressources informatiques, sont évaluées face aux outils existants.

Dans le chapitre 4, la qualité des distances calculées par Simka est évaluée sur des données réelles. Elles sont comparées aux mesures d'autres outils de comparaison *de novo*, à des indices traditionnels basés sur des compositions taxonomiques et à des résultats biologiques connus. Une étude du consortium Tara Oceans, dans laquelle nous avons été impliqués, est enfin exposée. Il s'agit d'une analyse comparative à l'échelle de la planète basée sur les résultats de Simka.

La chapitre 5 évalue l'impact du sous-échantillonnage de données sur les distances calculées par Simka. L'objectif est de mesurer la robustesse des distances et d'améliorer les performances globales. Le sous-échantillonnage est tout d'abord effectué au niveau des lectures, puis au niveau des k-mers. Ce travail a abouti à un nouvel outil permettant d'estimer rapidement les résultats de Simka.

Enfin, le chapitre 6 résume les contributions de cette thèse, puis énonce les pistes de recherche et d'approfondissement.

Chapitre 2

État de l'art de la métagénomique comparative *de novo*

Ce chapitre décrit l'état de l'art des méthodes de métagénomique comparative *de novo*. L'accent est mis essentiellement sur le traitement de jeux de données plein-génome qui offrent une meilleure résolution des communautés. Les méthodes présentées comparent les jeux de données sur la seule base de leur contenu génomique. Elles ne sont donc pas biaisées par l'inconsistance et l'incomplétude des banques de références. Elles n'ont pas non plus recours à une phase d'estimation de la diversité *de novo* (assemblage métagénomique ou *binning*). Cette phase est complexe et peut éliminer une quantité importante de lectures.

La section 2.1 rappelle les objectifs d'une méthode de métagénomique comparative *de novo* et présente les challenges auxquels elle est confrontée. Historiquement, les premières approches se basent sur des comparaisons de lectures. Elles sont présentées en section 2.2. Plus récemment, les nouvelles méthodes utilisent le k-mer (mot de taille k) comme seule unité de comparaison. La plupart des approches de cette catégorie a été publiée pendant cette thèse. Elles sont détaillées en section 2.3.

2.1 Objectifs et challenges

Soit un ensemble de N jeux de lectures métagénomiques, dénotés $S_1, S_2, S_i, ...S_N$. L'objectif est de fournir une matrice de similarité D de taille $N \times N$ où $D_{i,j}$ représente la similarité entre les jeux S_i et S_j (figure 2.1). Le calcul de la similarité fonctionne toujours par paire de jeux. La similarité est donnée par le pourcentage de lectures similaires. Deux séquences sont dites similaires si leur score d'alignement dépasse, par exemple, un certain seuil.

Deux principaux challenges algorithmiques découlent de cet objectif initial. Ils concernent tous les deux le passage à l'échelle. Le premier vient de la comparaison de deux jeux. Un jeu plein-génome contient des centaines de millions de lectures qu'il faut comparer aux séquences de l'autre jeu. Le deuxième vient du fait qu'il faut calculer $O(N^2)$ mesures de similarité entre toutes les paires des N jeux d'entrée. La métagénomique comparative *de novo* doit donc faire face à une complexité quadratique en N. Pour rappel, le projet Tara Oceans [3] met actuellement à disposition des centaines de jeux contenant des centaines de millions de lectures chacun.

Les méthodes *de novo* de comparaison existantes ont été catégorisées en deux familles : celles basées sur la comparaison de lectures et celles basées sur la comparaison de k-mers. La première famille essaie de rester proche des jeux de données originaux en délivrant une similarité basée sur un nombre de lectures similaires entre les paires de jeux. Les approches de cette famille ont deux objectifs principaux : (1) réduire le nombre de comparaisons de lectures à

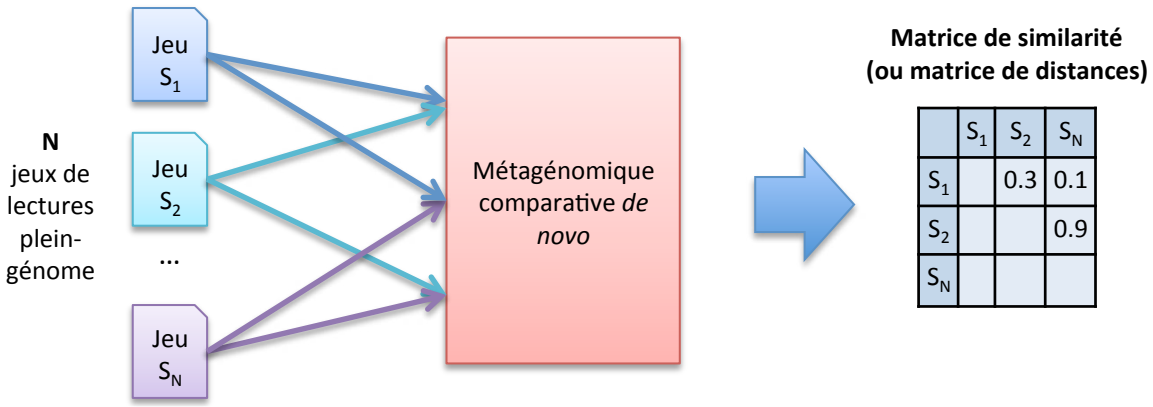

Figure 2.1. Aperçu d'une méthode de métagénomique comparative *de novo*. Une technique de métagénomique comparative *de novo* est utilisée entre chaque paire de jeux de lectures en s'appuyant seulement sur leur contenu génomique. Le résultat est une matrice de similarité (communément appelée matrice de distances) symétrique de taille $N \times N$ où chaque valeur donne la similarité entre une paire de jeux.

effectuer entre chaque paire de jeux ; (2) améliorer le processus de comparaison de deux lectures qui est traditionnellement effectué grâce à des techniques d'alignement coûteuses. La seconde famille oublie cette structure en lectures et voit les jeux de données comme des ensembles de k-mers. À l'inverse des lectures, les k-mers sont très rapides à traiter car leur comparaison se fait de manière exacte : deux k-mers sont égaux ou non. Cette approche découle du recul que l'on a sur l'utilisation du k-mer en bioinformatique des séquences, telle que sa probabilité d'être spécifique à un génome ou encore le fait que son abondance est proportionnelle à celle du génome d'où il provient. Cependant, pour valider ces observations, les k-mers doivent être suffisamment longs ($k \approx 21$) [65, 66]. Il est alors complexe de les manipuler car l'espace des k-mers (4^k) croît de manière exponentielle avec k. Dans un seul jeu de données de Tara Oceans, nous observons plusieurs milliards de 21-mers distincts. Ainsi, les challenges des méthodes de cette famille consistent à manipuler et à comparer un tel volume de k-mers.

Définitions Avant de présenter les techniques de métagénomique comparative *de novo*, nous définissons quelques notions centrales.

1. **Séquence.** Une séquence est composée de zéro ou plus de symboles appartenant à un alpabet Σ. Ici, le travail porte sur l'ADN. L'alphabet est alors $\Sigma = \{\text{A, C, G, T}\}$. Une séquence s de longueur n sur Σ peut alors se définir comme $s[0]s[1]s[2]...s[n-1]$ avec $s[i] \in \Sigma$ pour $0 \leq i < n$. La longueur d'une séquence s est représentée par $|s|$.
2. ***k*-mer.** On note $s[i, j]$ la sous-séquence $s[i]s[i+1]...s[j]$ de s. On dit alors que cette sous-séquence de s apparait à la position i. On appelle k-mer une séquence de longueur k et $s[i, i+k-1]$ est un k-mer apparaissant à la position i.
3. **Représentation des *k*-mers.** En pratique, chaque k-mer peut être représenté par un entier unique en codant chaque nucléotide sur deux bits : par exemple, A est codé par 00, C par 01, G par 10 et T part 11. Un k-mer correspond donc à une séquence de $2k$ bits. Par exemple, si $k \leq 32$, un k-mer peut être représenté par un entier natif de 64 bits. Avec cette représentation, leur indexation et comparaison est donc être extrêmement efficace.

2.2 Comparaison de métagénomes basées sur des comparaisons de lectures

Une idée simple pour mesurer la similarité entre deux jeux de lectures consiste à compter leur nombre de lectures similaires. La difficulté provient de cette notion de lectures similaires. Il ne s'agit pas simplement de retrouver les lectures parfaitement identiques mais celles qui partagent un chevauchement suffisamment long (par exemple, 70% de la taille des lectures). On émet alors l'hypothèse que ces deux séquences proviennent d'un même organisme. Il doit également être possible d'autoriser des mutations dans ce chevauchement qui peuvent être dues à des erreurs de séquençage ou à des variants génétiques des individus d'une même espèce. L'alignement de deux lectures peut être un processus très coûteux en temps. Rappelons que le problème initial consiste à effectuer cette action un nombre de fois quadratique pour comparer toutes les séquences entre elles.

Deux types d'approches ont été mises en œuvre pour effectuer de telles comparaisons. La première répond au problème de manière très précise avec des outils classiques d'alignement de séquences tels que Blast [8] et Blat [103]. La deuxième catégorie accélère fortement les calculs grâce à des méthodes de comparaison de séquences sans-alignement (*alignment-free*) et à de nouvelles techniques d'indexation.

2.2.1 Avec alignement de séquences

L'alignement de séquences est un problème classique de bioinformatique. Il s'agit de comparer deux séquences en autorisant divers types de mutations : des substitutions, insertions ou délétions de nucléotides. Les algorithmes d'alignement, tels que *Needleman–Wunsch* [7], utilisent la programmation dynamique pour trouver l'alignement optimal entre deux séquences et retournent un score de similarité. Ce score prend en compte le nombre de nucléotides en commun et les diverses mutations intégrées. Il existe deux types d'alignement : global et local. L'algorithme *Needleman–Wunsch* effectue des alignements globaux en tentant d'aligner les séquences sur toute leur longueur. L'algorithme *Smith–Waterman* [104] appartient à la deuxième catégorie et recherche les meilleurs alignements de différents segments des deux séquences. Ces régions peuvent être séparées et dans n'importe quel ordre. L'alignement local correspond à notre objectif de recherche de chevauchements entre les séquences. Ces deux algorithmes exacts possèdent une complexité en temps et en mémoire de $O(m \times n)$ où m et n sont les tailles des séquences. Ils ne passent pas à l'échelle des jeux de données actuels.

La plus répandue des heuristiques d'alignement se nomme ancrage et extension (*seed-and-extend*). L'étape d'ancrage consiste à identifier des segments de taille k (k-mers) parfaitement égaux. On dit que deux séquences se sont ancrées si elles partagent au moins un k-mer. L'étape d'extension étend les zones ancrées vers la gauche et vers la droite en utilisant des algorithmes d'alignement classique. Cette heuristique utilisée telle quelle ne résout pas notre problème de passage à l'échelle. Elle est populaire car elle peut être couplée à une étape d'indexation qui va réduire le nombre quadratique d'opérations à effectuer. Les outils d'alignement local, tels que Blast [8], commencent par indexer les k-mers d'un des deux jeux de lectures. Cet index permet de requêter en temps constant à quelles positions apparait un k-mer dans le jeu. Le second jeu, non indexé, est ensuite parcouru. Chaque lecture est découpée en k-mers afin de requêter l'index et d'identifier en temps constant les régions ancrables. Les lectures non ancrables sont ignorées et ne génèrent aucune opération d'alignement coûteuse.

Il existe aujourd'hui de nombreux outils "Blast-like". Ceux-ci font un compromis entre rapidité et qualité des résultats. La taille des k-mers d'ancrage est un facteur décisif. Plus celle-ci est grande, moins il y a d'alignements à effectuer mais moins l'outil est sensible. De manière

similaire, BLAT [103] est 500 fois plus rapide que BLAST mais a été élaboré pour trouver des alignements possédant au moins 95% d'identité. Ces outils restent malgré tout très lents pour comparer un grand nombre de séquences. Par exemple, l'auteur de BLAT estime qu'il faudrait 12 jours pour comparer deux jeux de données NGS de souris en utilisant 100 cœurs [103].

En métagénomique comparative *de novo*, on cherche à savoir si les séquences sont similaires, c'est à dire si leur taux d'identité est élevée. Par exemple, dans l'étude comparative du projet Global Ocean Sampling (GOS) [100], deux lectures ont été dites similaires si elles partagent un chevauchement dont le taux d'identité est d'au moins 90%. De telles séquences ont une ou plusieurs régions parfaitement communes. Des approches moins sensibles mais plus rapides pourraient être utilisées pour détecter de tels chevauchements. Il existe justement tout un pan de la littérature qui regroupe des méthodes, dites sans-alignement (*alignement-free*), pour comparer les séquences sans avoir recours aux alignements.

2.2.2 Sans alignement de séquences

La comparaison de séquences sans-alignement (*alignement-free*) est un domaine à part entière qui date des années 80 [36] dont la popularité a particulièrement augmenté ces dernières années tant la quantité de données générée en bioinformatique est importante. Le moyen classique pour déterminer la similarité entre deux séquences sans alignement consiste à calculer le pourcentage de k-mers partagés. Si ce pourcentage dépasse un certain seuil, les séquences sont dites similaires. Dans le champ de la métagénomique, quelques outils utilisant ce principe ont été développés. TRIAGETOOLS [105] a été créé initialement pour tester la présence d'un ensemble de séquences dans un jeu métagénomique. COMPAREADS [106] a été conçu spécifiquement pour comparer efficacement deux grands jeux métagénomiques. COMMET [107] est une extension de COMPAREADS qui le rend plus efficace pour comparer $N \times N$ jeux de données ($N > 2$). Pour éviter d'avoir à comparer toutes les séquences contre toutes, ces outils possèdent une méthodologie similaire à celle des approches de type BLAST présentées précédemment : les k-mers d'un des deux jeux sont indexés afin de ne considérer que les paires de séquences qui partagent au moins un k-mer.

TriageTools. TRIAGETOOLS indexe la présence-absence des k-mers d'un premier jeu de lecture, dit cible, grâce à un tableau de bits de taille 4^k, qui est le nombre total de k-mers distincts. La position d'un k-mer dans ce tableau est obtenue via sa représentation en entier (voir section 2.1). Tous les bits sont initialement à 0, puis mis à 1 si le k-mer est présent dans le jeu à indexer. Pour chaque lecture du second jeu non indexé, dit requête, son nombre de k-mers partagés avec les séquences du jeu cible est déterminé via l'index. Si ce nombre dépasse un certain seuil, alors cette lecture est dite similaire. L'avantage de cet index est sa rapidité de construction et d'accès puisque c'est un simple tableau. Il est également plus compact que l'index utilisé par les outils d'alignement qui stockent les positions où apparaissent chaque k-mer. Ce gain en espace est au prix de résultats moins précis. En effet, une lecture du jeu requête peut être dite similaire à tort car elle peut partager des k-mers provenant de différentes lectures du jeu cible. On appelle cela des faux-positifs. La figure 2.2 illustre le fonctionnement de TRIAGETOOLS et le concept de faux-positifs.

Les auteurs de TRIAGETOOLS ont pu estimer la probabilité de ces faux-positifs. Un des facteurs non négligeables est le taux de remplissage de l'index. La taille k doit donc être adaptée selon la diversité des métagénomes qui influence le nombre de k-mers distincts observés. Malheureusement, la mémoire utilisée par l'index de TRIAGETOOLS croit exponentiellement avec k. Celui-ci ne permet pas l'usage de k-mers plus grands qu'une quinzaine de nucléotides. Un autre inconvénient est que le calcul du nombre de séquences similaires est asymétrique. C'est

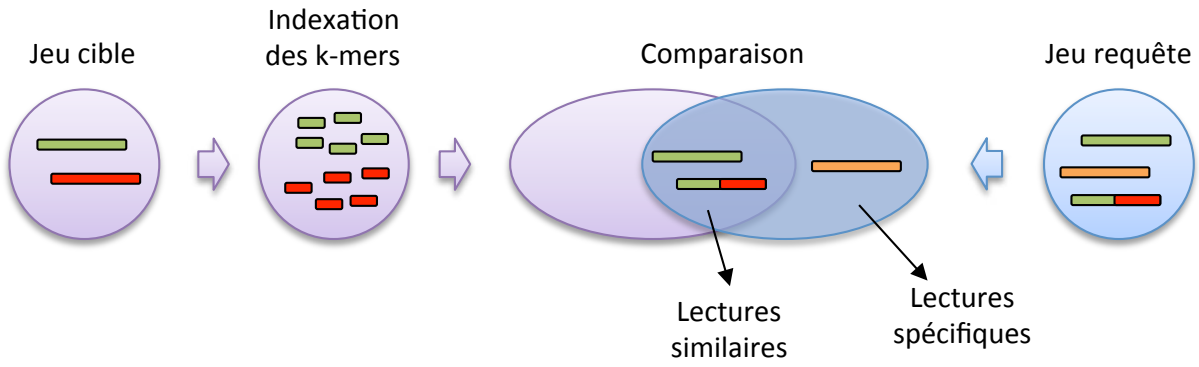

Figure 2.2. Processus de comparaison de deux jeux de lectures de TriageTools. Les grands rectangles représentent des lectures et les petits des k-mers. La couleur des séquences représente leur similarité. Les deux lectures vertes sont les seules à être véritablement similaires. Le jeu cible est indexé afin de tester en temps constant si les séquences du jeu requête partagent des k-mers avec les séquences du jeu cible. La séquence orange est bien identifiée dissimilaire car elle ne partage pas de k-mers avec le jeu cible. La séquence verte et rouge est dite similaire mais représente un faux-positif. Tous ses k-mers sont partagés mais proviennent de différentes lectures du jeu cible.

à dire que TriageTools détermine les séquences du jeu requête similaires au jeu cible mais pas l'inverse. Pour obtenir un pourcentage de séquences similaires symétriques, il faut également connaitre le nombre de lectures du jeu cible similaires au jeu requête. Il faut donc relancer l'outil en inversant le jeu qui est indexé et celui qui est requêté.

Compareads. Contrairement à TriageTools, Compareads [106] est un outil dédié à la métagénomique comparative et répond aux limitations de ce dernier, à savoir la gestion d'une taille de k-mer plus grande tout en restant très compact en mémoire. Compareads n'indexe également que la présence-absence des k-mers d'un des deux jeux. Comme TriageTools, il peut donc identifier à tort des séquences similaires (faux-positifs) et génère des résultats asymétriques. Cependant, la méthodologie employée ici permet de traiter efficacement ces deux points.

Premièrement, l'index de Compareads est un filtre de Bloom qui permet le stockage de plus grands k-mers ($k > 30$) tout en ayant une empreinte mémoire très faible. Un filtre de Bloom peut être vu comme une table de hachage qui ne gère pas les collisions des éléments que l'on y insère. Cela implique qu'un k-mer peut être identifié à tort comme présent dans le filtre. Cet index est donc une seconde source de faux-positifs. Cependant, la force du filtre de Bloom est que son taux de faux-positifs peut être maintenu très bas tout en restant très compact en mémoire. Pour environ 11 bits par k-mer, celui-ci génère moins de 0.1% de faux-positifs. Ces 11 bits représentent presque six fois moins de mémoire nécessaire au stockage de 30-mers normalement (représentation en entier).

Deuxièmement, Compareads propose une solution pour comparer plus rapidement les deux jeux dans les deux sens tout en réduisant le nombre de faux-positifs. Soit une paire de jeux de lectures S_1 et S_2, la comparaison de S_1 (requête) à S_2 (cible) implique l'indexation des k-mers de S_2, puis le parcours des lectures de S_1 pour déterminer les séquences de S_1 similaires à S_2. Ce processus de comparaison de base est le même que celui employé par TriageTools (figure 2.2). Les séquences de S_1 similaires à S_2 sont stockées dans un ensemble noté $S_{1>2}$. L'opération inverse doit ensuite être effectuée. Plutôt que de comparer S_2 à S_1, Compareads compare S_2 à $S_{1>2}$. Cela aboutit bien au même résultat mais $S_{1>2}$ contient beaucoup moins de séquences que S_1. Cela permet au passage d'éliminer des faux-positifs qui auraient pu être

générés par des k-mers des lectures en plus de S_1. Le résultat est stocké dans un ensemble $S_{2>1}$. Enfin, une troisième passe de comparaison $S_{1>2}$ avec $S_{2>1}$ est effectuée. Cela représente le même résultat que la comparaison de S_1 et S_2 mais avec moins de faux-positifs. CompareADS traite également chaque lecture plus rapidement que TriageTools. Une lecture est dite similaire si elle partage au moins t k-mers non chevauchants avec le jeu indexé. Ainsi, lorsqu'un k-mer est trouvé dans l'index, les k k-mers suivants sont passés sans générer d'accès à l'index.

Plusieurs tests de validation ont été effectués sur les données du projet GOS comportant 44 échantillons d'eau de mer. Dans un premier test, CompareADS et Blast ont été lancés sur une dizaine de jeux pour comparer le nombre de lectures qu'ils identifient similaires. La classification de ces jeux, sous la forme de dendrogrammes, a également été comparée qualitativement. Pour de faibles valeurs du seuil de k-mers partagés t ($t = 1$ et $t = 4$), le nombre de séquences identifiées similaires par Blast et CompareADS est sensiblement le même, mais il est très différent pour un seuil t élevé ($t = 10$). Cependant, dans tous les cas, les mêmes classifications d'échantillons sont obtenues. Ce test est important car il montre qu'il n'est pas forcément nécessaire de considérer la valeur absolue d'une mesure de similarité mais plutôt de la voir comme une mesure relative destinée à être comparée aux autres mesures. Dans un second test, CompareADS a été lancé sur tous les jeux de données du projet GOS. Comme dans l'étude originale, qui a utilisé une approche avec alignement, CompareADS a regroupé les échantillons ayant une position géographique proche et a donc abouti encore une fois aux mêmes conclusions.

La complexité en temps de CompareADS est linéaire sur le nombre de k-mers des deux jeux. Celui-ci parvient à comparer deux grands jeux de Tara Oceans en une dizaine d'heures. Blast aurait demandé des années pour parvenir à un résultat similaire. Un des inconvénients de CompareADS est qu'il génère des jeux de lectures intermédiaires sur le disque (ensembles $S_{1>2}$ et $S_{2>1}$) qui peuvent devenir volumineux sur de gros jeux de données. De plus, CompareADS n'est pas prévu pour comparer plus de deux jeux. Si N jeux sont disponibles ($N > 2$), celui-ci doit être relancé $O(N^2)$ fois pour obtenir une matrice de similarité. Cette approche est très inefficace car elle effectue énormément d'opérations redondantes. Notamment, un même jeu est lu et indexé N fois.

Commet. Commet est une extension de CompareADS pour optimiser la comparaison de N jeux de données ($N > 2$). L'idée de Commet est de comparer $1 \times N$ jeux de lectures à la fois. C'est à dire qu'une fois qu'un jeu est indexé, cet index est utilisé contre les lectures des $N-1$ jeux restants sans le reconstruire. La phase d'indexation se produit donc N fois plutôt que N^2 fois. La seconde innovation de Commet est une représentation efficace des jeux de lectures intermédiaires. Par exemple, pour stocker l'ensemble $S_{1>2}$ (les séquences de S_1 similaires à S_2), un tableau de bits $B_{1>2}$ est utilisé où le bit à la position i indique si la $i$ème lecture de S_1 est similaire à S_2 ou non. L'ensemble des lectures de $S_{1>2}$ est reconstruit en lisant S_1 tout en vérifiant la valeur des bits du tableau $B_{1>2}$. Cette représentation offre un taux de compression très important puisqu'une lecture entière est résumée à 1 bit. Dans les expérimentations de la publication de Commet, celui-ci parvient à être 2 à 3 fois plus rapide que CompareADS et génère des fichiers intermédiaires 1000 fois plus petits.

Cependant, ces optimisations n'offrent pas un passage à l'échelle considérable. Pour obtenir les N^2 mesures de similarité, chaque jeu doit toujours être lu entièrement $3N$ fois, sans compter la phase d'indexation. Ce nombre peut même être décuplé sur de gros jeux de données à cause de la taille maximale autorisée pour le filtre de Bloom servant d'index (4 Go). Lorsque l'index du jeu cible est jugé rempli, à raison d'une dizaine de bits par k-mer distinct inséré, Commet enclenche la comparaison du jeu partiellement indexé contre les $N-1$ autres. Cette opération est répétée tant qu'il reste des k-mers distincts du jeu cible à indexer. Un jeu de Tara Oceans peut largement faire déborder la taille de cet index et provoquer une combinatoire très importante

impliquant notamment un grand nombre d'accès disque.

L'outil Commet est la dernière grande avancée concernant la comparaison de métagénomes basée sur la comparaison de leurs lectures. Cette approche ne permet malheureusement pas de passer à l'échelle sur un grand ensemble de grands jeux métagénomiques. En effet, Commet requiert environ 10h pour comparer deux jeux de Tara Oceans et le nombre de mesures de similarité à calculer augmente quadratiquement avec le nombre de jeux de lectures. Cependant, la validation de ces approches sans-alignement ont permis de mettre en valeur la force du k-mer comme unité de comparaison. Cela a initié une nouvelle vague de méthodes beaucoup plus performantes basées sur la seule information de la composition en k-mers des jeux de lectures.

2.3 Comparaison de métagénomes basées sur des comparaisons de k-mers

L'avantage des k-mers est que leur comparaison se fait de manière exacte : ils sont égaux si leur séquence est exactement la même et différents sinon. En pratique, les k-mers sont même vus comme des entiers (voir section 2.1), ce qui rend leur indexation et leur comparaison extrêmement rapide. Les outils récents, qui ne se basent que sur l'information de la composition en k-mers des jeux, peuvent ainsi passer à l'échelle sur la comparaison de centaines de métagénomes.

Intuitivement, la perte de la structuration originale des jeux de lectures laisse penser que ce gain de performances est au prix d'une forte dégradation de la précision des résultats de comparaisons. En effet, l'usage des k-mers ne vient pas d'une motivation biologique mais d'un procédé informatique pour passer à l'échelle. Cependant, nous possédons aujourd'hui un certain recul par rapport à cet usage et certaines observations montrent que le k-mer est une unité pertinente pour comparer des communautés : (1) les k-mers suffisamment longs ($k > 15$) sont souvent spécifiques d'un génome [65], (2) l'abondance des k-mers est proportionnelle à l'abondance des génomes [66] et (3) deux organismes proches possèdent des compositions en k-mers plus proches que deux organismes éloignés [70]. L'intérêt du k-mer a également été montré par les outils de comparaison sans-alignement, tel que Compareads, qui voient très clairement un des deux jeux de lectures à comparer comme un ensemble de k-mers. Comme vu précédemment pendant la description de ces approches, la complexité de la gestion des k-mers en termes de stockage et de comparaisons vient de leur très grand nombre dans la gamme de valeurs de k qui nous intéresse (autour de 15 à 30 au moins). Par exemple, un échantillon de Tara Oceans d'environ 100 millions de lectures contient en moyenne plus de 6 milliards de 31-mers distincts. Cette richesse, en termes de k-mers distincts, provient de la complexité de chaque génome présent dans les échantillons ainsi que des erreurs de séquençage qui peuvent générer chacune jusqu'à k nouveaux k-mers distincts erronés.

Les approches de comparaison basées sur les k-mers peuvent être séparées en deux familles. La première voit un jeu de lectures comme un ensemble de k-mers et ne prend donc en compte que leur présence-absence dans sa mesure de similarité. La deuxième voit un jeu comme un multiensemble (ou sac) de k-mers et prend donc en compte leur nombre d'occurrences. Le problème est beaucoup plus complexe dans le deuxième cas car il faut compter les k-mers. Lorsque k est grand, cette tâche est difficile. C'est un des problèmes fondamentaux du traitement de données NGS. Il est important de noter que la deuxième famille peut fournir les résultats de la première en mettant tous les comptages des k-mers présents à 1. Le problème de comparaison basé uniquement sur la présence-absence des k-mers a été clos récemment par Mash [108]. La comparaison prenant en compte l'abondance des k-mers est quant à elle un sujet très actif. Plusieurs outils s'attaquent au problème, incluant celui qui a été développé pendant cette thèse

et qui sera présenté en détails dans le chapitre suivant.

2.3.1 Comparaison basée sur la présence-absence des *k*-mers

Un outil très récent, nommé MASH [108], a bousculé le paysage de la métagénomique comparative *de novo*. MASH s'appuie sur la technique MINHASH [109] pour comparer les ensembles de k-mers. MINHASH est une approche statistique qui permet d'estimer l'index de Jaccard entre deux ensembles en n'utilisant que quelques milliers de leurs éléments. Cette technique a une emprise très forte actuellement en bioinformatique car on retrouve le calcul de l'index de Jaccard dans de nombreuses applications, la plus connue étant la comparaison de deux séquences basée sur leurs k-mers. Il est donc intéressant de comprendre le cœur de son fonctionnement. De plus, une des contributions de cette thèse, présentée dans le chapitre 5, est une adaptation de MINHASH.

Considérons les deux ensembles suivants :

$S_1 = \{\text{A, } \mathbf{F}\text{, C, D, } \mathbf{E}\text{, B, } \mathbf{G}\}$

$S_2 = \{\text{I, } \mathbf{E}\text{, } \mathbf{G}\text{, H, } \mathbf{F}\text{, J}\}$

Ils se composent de 10 éléments distincts dont 3 partagés (indiqués en gras). Par conséquent, leur index de Jaccard est de 3/10. Théoriquement, le processus de MINHASH consiste à prendre l'union des deux ensembles, à le mélanger aléatoirement, et à sélectionner sa première valeur. Ainsi, si l'union de notre exemple est considéré :

$S_1 \cup S_2 = \{\text{A, } \mathbf{F}\text{, C, D, } \mathbf{E}\text{, B, } \mathbf{G}\text{, I, H, J}\}$

et qu'il est mélangé, la probabilité que son premier élément soit un élément partagé est de 3/10, la même que l'index de Jaccard. Si un seul mélange est effectué, l'estimation de l'index peut valoir 0, le premier élément est spécifique, ou 1, il est partagé. Pour obtenir une meilleur estimation, n permutations sont effectuées en vérifiant à chaque fois si le premier élément est partagé ou spécifique. L'index de Jaccard estimé J' est alors donné par la moyenne de 1 obtenue :

$$J'(S_1, S_2) = \frac{p}{n}$$

où p est le nombre d'éléments partagés. L'erreur espérée de cette estimation est en $O(1/\sqrt{n})$.

La méthode MINHASH permet de faire ces permutations de l'union et la détection des éléments partagés de manière très efficace. En théorie, cette tâche n'est pas triviale. Les ensembles peuvent être très grands, voire être des flux de données dont on ne connait pas la taille *a priori*. La clé de MINHASH est une fonction de hachage $h(x)$ qui prend des entiers x en entrée et les transforme en d'autres entiers, sans collision. Par exemple, imaginons que h prenne et retourne des entiers de 32 bits. Appliquer h à tous les éléments d'une liste contenant tous les entiers de 0 à $2^{32} - 1$ permet d'obtenir une nouvelle liste avec les mêmes nombres dans un ordre aléatoire. La fonction $h_{min}(S_i)$ de MINHASH consiste à hacher tous les éléments d'un ensemble S_i et à mémoriser le plus petit. Il s'agit de notre estimateur qui peut retourner 1 si $h_{min}(S_1)$ et $h_{min}(S_2)$ sont égaux et 0 sinon. Pour améliorer l'estimation, n fonctions $h_{min}(S_i)$ sont calculées par ensemble en utilisant n fonctions de hachage différentes. Les n résultats sont insérés dans une liste qu'on appelle "signature". Les mêmes fonctions de hachage sont utilisées pour déterminer la signature de chaque ensemble. L'estimation $J'(S_1, S_2)$ est simplement calculée en comptant le nombre d'éléments partagés par leur signature. Le calcul d'une signature n'est à faire qu'une fois, il est indépendant des autres jeux.

MASH emploie une variante de MINHASH plus simple d'un point de vue calculatoire. Une seule fonction de hachage est utilisée pour hacher tous les k-mers d'un jeu. Une liste des n plus petits k-mers hachés est maintenue. Ces éléments forment la signature MINHASH. Pour comparer

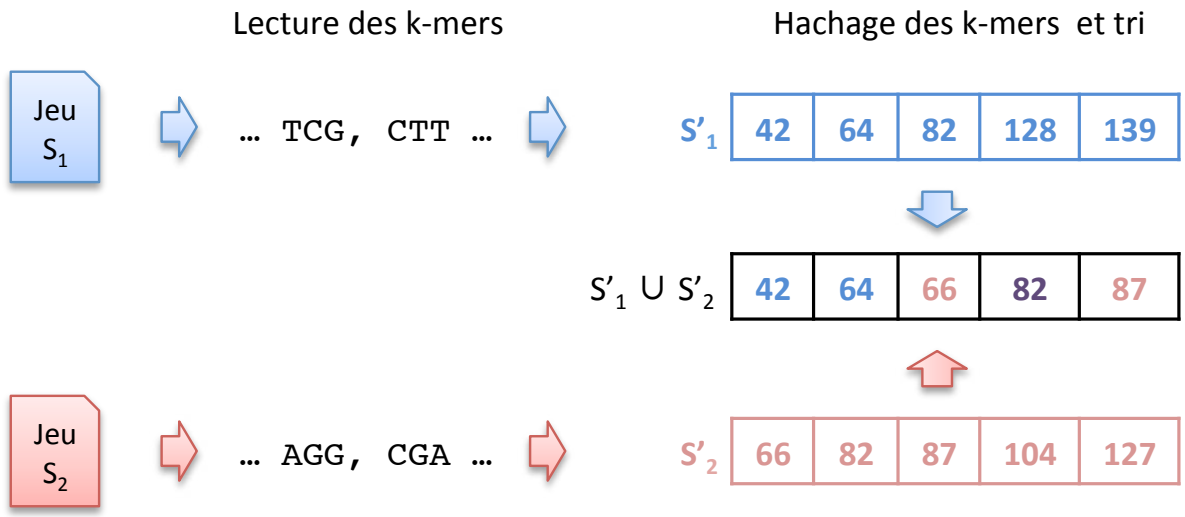

FIGURE 2.3. Processus de comparaison de MASH en utilisant la technique Minhash. Les k-mers des deux jeux sont extraits, puis hachés en entier. Les signatures MINHASH S'_1 et S'_2 de taille $n = 5$ représentent respectivement les n plus petits k-mers hachés de S_1 et S_2. La fusion de S'_1 et S'_2 produit l'ensemble $S'_1 \cup S'_2$ constitué de leurs n plus petites valeurs. Cet union est un sous-ensemble aléatoire de $S_1 \cup S_2$ dans lequel un élément partagé a été détecté pendant la fusion. L'estimation de l'index de Jaccard de S_1 et S_2 vaut donc $1/5$.

deux signatures, un algorithme de tri-fusion est effectué. La fusion s'arrête lorsque n éléments distincts ont été vus. Cet arrêt permet de s'assurer que si un élément est partagé et qu'il est pris en compte dans une signature, alors il sera vu dans l'autre. La figure 2.3 illustre le fonctionnement de cette variante de MINHASH. MASH utilise la fonction de hachage MurMurHash3 pour mapper uniformément les k-mers dans un ensemble d'entiers de 64 bits. La probabilité de collision entre deux k-mers est très proche de zéro.

En termes de performances, il semble peu probable de pouvoir être plus efficace que MASH. La complexité en temps pour obtenir la signature d'un jeu est en $O(m \log n)$ où m est le nombre de k-mers et n est la taille de la signature. De plus, au fur et à mesure que les k-mers sont traités, la plus grande valeur de la liste triée diminue. Par conséquent, la probabilité qu'un futur élément y soit inséré diminue également. Cet algorithme est donc quasiment linéaire quand $n << m$ [109]. Le calcul des signatures est indépendant et peut donc être lancé en parallèle. Le calcul des distances est extrêmement rapide compte tenu de la taille des signatures. La complexité quadratique de la phase de comparaison est donc bien gérée par MASH. Sur un important projet constitué de 888 jeux de données provenant des projets HMP [1] et MetaHIT [24] (~40 milliards de lectures), MASH n'a requis que 7h de traitement sur une machine équipée de 40 cœurs. Son point faible se situe au niveau des résultats fournis qui se limitent à une estimation de l'index de Jaccard. Cette mesure ne prend en compte que la présence-absence des k-mers et non leur nombre d'occurrences. Toute la famille d'indices de comparaison quantitatifs, telle que la très populaire dissimilarité de Bray-Curtis [87], n'est ainsi pas fournie par MASH.

2.3.2 Comparaison basée sur le comptage des *k*-mers

Les méthodes de comparaisons basées sur des comptages de k-mers ont un fonctionnement semblable à celui des approches classiques basées sur la diversité taxonomique. Ici, les comptages d'espèces sont simplement remplacés par des comptages de k-mers. La différence majeure est la dimension des données à manipuler et à comparer lorsque de grands k-mers sont considérés. Le

fonctionnement de cette approche est le suivant :

1. **Comptage des *k*-mers.** Le spectre de k-mers de chacun des N jeux de données est calculé indépendamment. Un spectre de k-mers est l'ensemble des k-mers distincts d'un jeu de données, chacun associé à leur abondance dans ce jeu.
2. **Comparaison des spectres.** Les spectres sont comparés par paire afin d'en dériver des indices de similarité basés sur leur nombre de k-mers partagés et spécifiques.

Le comptage de grands k-mers est loin d'être un problème trivial. Un aperçu des challenges de ce domaine et des techniques existantes est présenté dans la sous-section suivante. Une des contributions de cette thèse exploite la méthodologie de ces outils. Dans un second temps, nous abordons les méthodes de comparaison de métagénomes qui exploitent ces spectres.

2.3.2.1 Comptage des *k*-mers

À première vue, compter les mots dans un texte est simple. Pour cela, une manière de procéder est d'utiliser une structure de données en mémoire de type dictionnaire associant chaque mot distinct à son comptage. En bioinformatique des séquences, ce problème est pourtant un domaine de recherche à part entière dû au très grand nombre de k-mers distincts qui peuvent être insérés dans ce dictionnaire lorsque k est grand ($k > 15$). Par exemple, à raison de 8 octets par 31-mers distincts et 4 octets pour leur comptage, une approche par dictionnaire requiert déjà plus de 72 Go de mémoire sur un jeu de Tara Oceans. Cette approche est également difficilement parallélisable.

Cette opération est fondamentale en bioinformatique car elle permet de détecter (en grande partie) les k-mers possédant une erreur de séquençage. Puisque qu'un caractère aléatoire a été introduit dans leur séquence, ces k-mers erronés vont avoir un comptage faible alors que les k-mers génomiques (non erronés) auront une abondance proche de la couverture du génome séquencé. Les erreurs de séquençage sont d'ailleurs une des sources de difficulté du problème. Chaque erreur de séquençage peut générer jusqu'à k k-mers distincts erronés. Ainsi, plus un jeu est grand, plus il y a d'erreurs de séquençage et plus il y a de k-mers distincts. Par exemple, l'équipe qui a séquencé et assemblé le génome du panda géant [110] (couverture de 56×) a dénombré 8.62 milliards de 27-mers distincts parmi lesquels 68% étaient rares et ont été filtrés.

Il existe ainsi plusieurs outils de comptage de k-mers qui peuvent être distingués en deux familles : ceux basés sur la mémoire et ceux basés sur le disque. Jellyfish [111] fait partie de la première famille, il optimise l'approche par dictionnaire mais son empreinte mémoire augmente avec la taille des jeux de données à traiter. DSK [112] compte les k-mers avec une mémoire constante en exploitant le grand réservoir qu'est le disque. Cette approche permet de compter les k-mers de très grands jeux de lectures sur des machines modestes. KMC2 [113] est la dernière grande avancée en termes de comptage de k-mers. Celui-ci réduit notamment drastiquement l'empreinte disque de DSK. Les indications de performance des outils sont issues de la publication de KMC2 [113]. Les tests ont été effectués sur un jeu de données génomiques humain contenant un milliard de lectures.

Jellyfish. Jellyfish [111] est un des compteurs de k-mers le plus populaire en bioinformatique car il a longtemps été le plus rapide. La principale innovation de Jellyfish est l'implémentation d'un dictionnaire (ou table de hachage associative) pouvant être modifié en parallèle sans utilisation de mutex pour gérer la synchronisation des différents threads. Malgré des efforts pour compresser les k-mers insérés et leurs comptages, l'espace mémoire requis par Jellyfish reste linéaire par rapport au nombre de k-mers distincts. Ainsi, sur de grands jeux de données, Jellyfish dépasse généralement les capacités d'un ordinateur standard. Par exemple, Jellyfish

demande 62 Go de mémoire pour compter les 28-mers du jeu humain et plus de 128 Go pour compter ses 55-mers.

DSK. Contrairement à la mémoire vive, le disque est une ressource dont nous disposons généralement en abondance. De plus, la technologie SSD rend les accès disque de plus en plus rapides. DSK [112] est le premier outil à avoir eu l'idée d'utiliser cet espace pour réduire l'empreinte mémoire du comptage des k-mers. L'idée est de répartir les k-mers dans P fichiers écrits sur le disque. On appellera ces fichiers de k-mers des partitions. Les P partitions sont des sous-ensembles disjoints de k-mers. Chacune d'elles peut donc être comptée indépendamment avec une empreinte mémoire plus faible que si tous les k-mers avaient été considérés en même temps. Cette caractéristique permet également la parallélisation du comptage.

Pour répartir les k-mers en P partitions disjointes et de taille similaire, ils sont hachés, puis transformés dans l'intervalle $[\![0, P[\![$ grâce à une opération *modulo*, ce qui donne la fonction de répartition suivante :

$$p(w) = h(w)\%P$$

où $p(w)$ est le numéro de partition du k-mer w et $h(w)$ est une fonction de hachage appliquée à la séquence du k-mer w. Cette fonction envoie bien les multiples occurrences d'un même k-mer dans la même partition.

DSK compte chaque partition avec une approche standard par dictionnaire. À la fin de ce processus, les P spectres sont concaténés afin de former le spectre de k-mers final. Un des principaux avantages de cette approche est que son empreinte mémoire est constante et peut être spécifiée par l'utilisateur. DSK estime alors le nombre de k-mers du jeu en analysant un échantillon de ses lectures et ajuste le nombre de partitions en conséquence. Le nombre de cœurs de calcul à disposition est également pris en compte dans l'équation pour optimiser la parallélisation. Sur le jeu de données humain, DSK est environ 6 fois plus lent que Jellyfish pour compter des 28-mers, mais ne demande que 6 Go de mémoire contrairement aux 62 Go de Jellyfish. La lenteur de DSK provient principalement de la phase de partitionnement des k-mers qui duplique la taille originale du jeu de données sur le disque. En effet, sur ce même jeu, l'empreinte disque de DSK monte à 263 Go pour stocker les partitions de k-mers. Cette forte empreinte disque est le goulot d'étranglement de DSK.

KMC2. La phase d'écriture de tous les k-mers sur le disque coûte cher en temps de calcul et semble inefficace d'un point de vue entropique. En effet, il existe une forte redondance (de taille $k-1$) entre deux k-mers successifs. Chaque nouveau k-mer écrit sur le disque n'apporte qu'un nucléotide d'information par rapport au précédent. Cette redondance pourrait être attrapée si les k-mers successifs étaient envoyés dans la même partition. Il suffirait alors d'écrire la séquence contigüe de la lecture contenant ces k-mers sans la découper. Les méthodes de comptage de k-mers qui ont suivi DSK, telles que KMC2, sont parties en quête de cette fonction de répartition qui enverraient les k-mers successifs d'une lecture dans la même partition.

Une telle fonction a été introduite plus tôt dans la littérature [114]. Dans cette publication, Roberts *et al.* montrent une technique pour réduire la quantité de k-mers à indexer par les méthodes d'alignement utilisant l'heuristique ancrage et extension. Cette technique sélectionne un représentant unique au sein d'une série de k-mers, appelé *minimizer*. Un minimizer est la plus petite séquence de taille m d'un k-mer au sens lexicographique ($m << k$). Lorsqu'on calcule le minimizer de tous les k-mers d'une lecture, on remarque que les k-mers successifs partagent très souvent le même minimizer (figure 2.4). Li et Yang [115] utilisent pour la première fois ce principe pour partitionner les k-mers. Ils introduisent la notion de super-k-mer qui est une

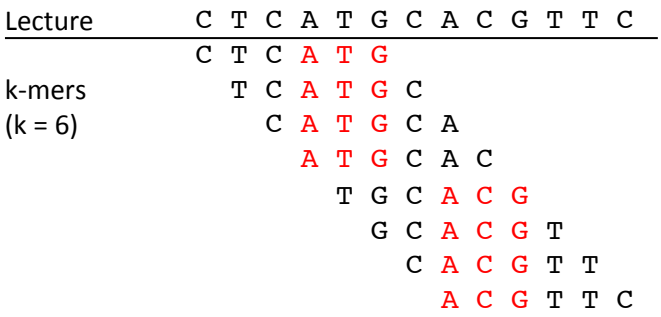

Figure 2.4. Représentation des *k*-mers d'une lecture et leur minimizer. Un minimizer (représenté en rouge) est le plus petit m-mer d'un k-mer au sens lexicographique ($m << k$). Dans cet exemple, $k = 6$ et $m = 3$. Cette lecture contient 2 minimizers et donc 2 super-k-mers : CTCATGCAC et TGCACGTTC.

région contigüe d'une lecture contenant des k-mers partageant le même minimizer. Ces super-k-mers sont écrits tels quels dans les partitions sans les découper en k-mers. Le numéro de la partition d'un super-k-mer est obtenu par l'intermédiaire de la séquence de son minimizer. Le principal défaut de cette fonction de répartition par minimizer est qu'elle génère des partitions de tailles dissimilaires puisque certains minimizers sont sur-représentés. Cela réduit l'intérêt d'une approche de comptage par disque puisque son empreinte mémoire est liée à la taille de la plus grande des partitions.

Les auteurs de KMC2 améliorent cette fonction de répartition en appliquant des règles sur les minimizers. Celles-ci consistent notamment à interdire les minimizers sur-représentés. De plus, KMC2 utilise une autre approche pour compter les k-mers : le tri-comptage (*sorting count*). Cet algorithme consiste à insérer tous les k-mers dans une liste et à la trier. Les k-mers distincts et leur comptage peuvent alors être identifiés en une passe sur la liste. La technique de tri utilisée exploite intensivement la capacité des processeurs modernes grâce aux instructions SIMD (Single Instruction Multiple Data) [116]. De plus, certaines sous-séquences des super-k-mers possédant certaines propriétés particulières peuvent être triées sans les découper en k-mers, ce qui réduit considérablement le nombre d'éléments à trier.

L'outil KMC2 est la dernière grande avancée en termes de comptage de k-mers. Son empreinte disque est quasiment trois fois inférieure à celle de DSK. De plus, KMC2 est plus rapide que Jellyfish tout en utilisant une quantité constante de mémoire. Cet outil permet le comptage des k-mers d'un grand jeu de données de Tara Oceans en une dizaine de minutes sur une machine modeste. Grâce à ces avancées, la comparaison de grands jeux métagénomiques basée sur leur comptage de k-mers est aujourd'hui envisageable.

2.3.2.2 Comparaison des spectres de *k*-mers

Les méthodes basées sur les spectres de k-mers pour comparer les métagénomes sont apparus à partir de 2014. Dubinkina *et al.* [117] montrent la pertinence de cette approche sur une petite plage de taille de k-mer ($k < 16$). DSM [118] est la première approche à parvenir à traiter un grand projet métagénomique en se basant sur de grands k-mers en un temps raisonnable. Cependant, celle-ci requiert un important cluster de calcul. Metafast [119] propose de ne considérer qu'un sous-ensemble de k-mers sélectionnés via une étape d'assemblage.

Comparaisons basées sur des petits *k*-mers. Dubinkina *et al.* [117] proposent d'évaluer la qualité de la comparaison par spectre de k-mers sur près de 300 jeux de données réels provenant

de deux études du microbiome intestinal humain [120, 121]. Ce travail se limite à de petits k-mers ($k < 16$). Cela permet de compter sans problème les k-mers de ces jeux en utilisant une matrice de comptage de taille $W \times N$ où W est la taille de l'espace des k-mers (4^k) et N est le nombre de jeux. À partir de cette matrice, la dissimilarité de Bray-Curtis entre chaque paire de jeux est calculée.

Pour mesurer la qualité de ces distances *de novo*, celles-ci sont comparées à des distances de Bray-Curtis taxonomiques. Ces distances taxonomiques sont calculées à partir de comptages d'espèces, déterminés en alignant les lectures sur un catalogue de génomes de référence. L'objectif est de voir s'il existe une relation entre les mesures *de novo* et les mesures taxonomiques. Dans un premier temps, l'allure du nuage de points est observée sur un graphique où l'axe X correspond aux distances taxonomiques et l'axe Y correspond aux distances basées sur les k-mers (figure 2.5). Dans un second temps, la relation est quantifiée. Pour cela, les paires de jeux sont triées par ordre de distance. À partir de deux ordonnancements, des mesures de corrélation peuvent être calculées, telle que la corrélation de Spearman. Cette mesure évalue les distances de manière relative, plutôt que de manière absolue ; seul l'ordre compte. La corrélation de Spearman renvoie une valeur entre -1 (relation anti-corrélée) et 1 (relation parfaitement corrélée). Une valeur de Spearman de 0 signifie qu'il n'existe aucune corrélation entre les deux mesures.

Cette étude montre tout d'abord une corrélation de Spearman assez faible de 0.44 entre les deux types de distances. Un grand nombre de valeurs aberrantes est visible sur le nuage de points (figure 2.5 gauche). Afin d'enquêter sur la présence de ces différences, l'impact des lectures non alignées sur les génomes de référence a été mesuré. Pour cela, une nouvelles matrice de distances *de novo* a été calculée en ne considérant que les lectures alignées. Cette manipulation fait effectivement disparaitre les valeurs aberrantes (figure 2.5 droite). La corrélation de Spearman est alors de 0.73. Cela montre clairement l'importance des méthodes *de novo* qui délivrent une information toute autre que les approches par références qui peuvent rejeter de nombreuses séquences non alignées.

Dans cet étude, il est montré que la corrélation augmente sur une plage de k allant de 5 à 12. Cependant, la méthodologie employée ne permet pas d'explorer de grandes tailles de k-mer. On ne sait pas si cette corrélation continue d'augmenter sur de plus grands k et s'il existe un optimum.

DSM. DSM [118] est le premier outil qui est parvenu à comparer des centaines de grands jeux en se basant sur de grands k-mers ($k > 21$). Sa méthode s'appuie sur un algorithme de tri-fusion pour comparer les grands spectres. Cela implique le tri de chaque spectre indépendamment par ordre lexicographique de leurs k-mers, puis leur fusion en parcourant les k-mers du plus petit au plus grand. Cette approche permet de reconstituer une matrice de comptage de taille $W \times N$, où W est le nombre de k-mers distincts parmi N jeux. DSM en extrait l'index de Jaccard entre les jeux ainsi que deux indices basés sur des distances euclidiennes qui prennent en compte l'abondance des k-mers.

Pour ne pas avoir à effectuer un tri explicite des grands spectres qui pourrait être très long, DSM emploie une méthode permettant de trier et de compter les k-mers en même temps. Cette approche se base sur l'arbre des suffixes [122]. Un arbre des suffixes est une structure de données arborescente utilisée pour indexer une séquence. Il est alors possible d'y mener un grand nombre d'opérations complexes efficacement comme tester si un k-mer apparait dans la séquence indexée, et même trouver son nombre d'occurrences. Un jeu de données peut être indexé de cette manière en concaténant toutes ses séquences séparées par un caractère spécial. La structure de cet arbre permet de lire ses k-mers du plus petit au plus grand par ordre lexicographique en partant de la racine vers les feuilles. Lorsqu'une séquence de taille k est atteinte dans l'arbre, il suffit de compter le nombre de feuilles parmi ses descendants pour obtenir son comptage. DSM construit

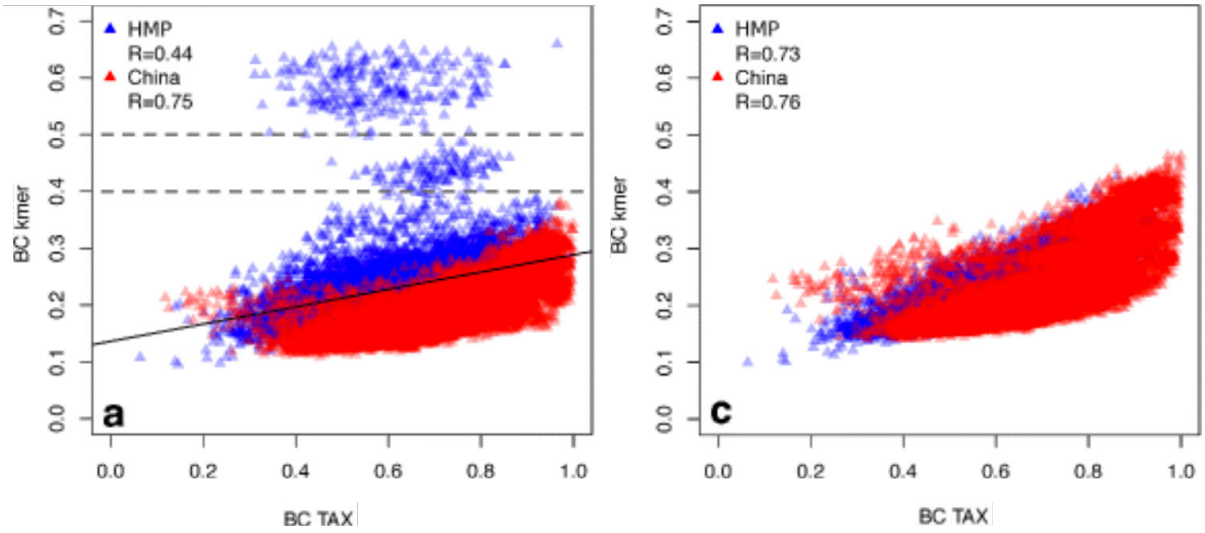

Figure 2.5. Corrélation entre des distances taxonomiques et des distances *de novo* basées sur des petits *k*-mers. Chaque point représente une paire d'échantillons. La coordonnée X indique la distance de taxonomique, et la coordonnée Y la distance *de novo* ($k = 11$). La valeur R est le coefficient de corrélation de Spearman entre la matrice de distances taxonomiques et celle *de novo*. Dans la figure de gauche, les distances *de novo* sont basées sur toutes les lectures des jeux de données, alors que sur celle de droite, seules les lectures alignées sur les références sont considérées. (figure extraite de Dubinkina *et al.* [117]).

un tel arbre par jeu de lectures, parcourt leurs k-mers par ordre lexicographique et les fusionne simultanément.

Sur le projet HMP (plus de 400 jeux contenant en moyenne 45 millions de lectures de 100 bp), DSM est parvenu à obtenir des matrices de distances basées sur la comparaison de 21-mers en un peu plus de 3 jours. Au moment de sa publication, l'outil de métagénomique comparative *de novo* le plus rapide, à savoir Commet, aurait requis plusieurs années de calcul. La faiblesse de DSM est son empreinte mémoire. Un arbre des suffixes a une complexité linéaire avec la taille de la séquence indexée. Malgré les efforts pour compresser cette structure [123], DSM requiert plus de 3 To de mémoire vive sur le projet HMP. Pour gérer de tels projets, DSM distribue la mémoire sur un large cluster de calcul. Cependant, cette quantité de mémoire étant liée aux nombres de k-mers des jeux, il n'est pas envisageable pour DSM de passer à l'échelle sur de plus grands projets métagénomiques.

Metafast. Metafast a un fonctionnement proche d'une méthode classique de comparaison basée sur des références. Metafast construit ses propres références via une étape d'assemblage grossière (pré-assemblage sans contigage) basée sur les k-mers. L'abondance des composantes assemblées obtenues est estimée en utilisant le comptage de k-mers les composant. Enfin, ces composantes sont utilisées pour comparer les métagénomes.

Pour générer ces assemblages, Metafast s'appuie sur une structure de données classique : le graphe de *de-Bruijn* [13]. Un graphe de *de-Bruijn* est un graphe dirigé où les nœuds sont des k-mers et une arête existe entre deux k-mers s'ils se chevauchent sur $k-1$ nucléotides. Un nœud est dit "simple" s'il possède exactement une entrée et une sortie et "branchant" sinon. La construction de ce graphe commence généralement par une phase de comptage des k-mers, afin de ne pas y insérer les k-mers peu représentés. Ces k-mers rares proviennent majoritairement d'erreurs de séquençage et complexifient inutilement le graphe en plus d'augmenter drastiquement le coût du stockage. Les spectres de k-mers sont écrits sur le disque car ils sont réutilisés

à chaque étape du processus de METAFAST.

METAFAST commence par construire un graphe de *de-Bruijn* pour chaque jeu de lectures indépendamment. Les chemins simples sont extraits de ce graphe. Un chemin simple, communément appelé unitig, est une série de nœuds simples connectés. Seuls les unitigs suffisamment longs sont conservés (> 100 *bp* par défaut). Les nœuds branchants sont également jetés. Les k-mers des unitigs de chaque jeu ayant passé ce premier filtre sont insérés dans un unique graphe de *de-Bruijn* afin de réaliser un assemblage croisé de tous les jeux de données. Les composantes connexes sont extraites de ce graphe. Seules celles contenant suffisamment de k-mers sont conservées.

Une matrice de comptage MC de taille $W \times N$ est ensuite construite, où W est le nombre de composantes connexes et N le nombre de jeux. $MC_{i,j}$ représente le nombre de fois où la composante i est présente dans le jeu S_j en nombre de k-mers. Pour cela, METAFAST charge successivement chaque spectre de k-mers en mémoire dans un dictionnaire et calcule l'abondance d'une composante comme le nombre total de k-mers appartenant à cette composante. Une matrice de dissimilarité de Bray-Curtis est finalement calculée à partir de la matrice MC.

Des tests de validation des performances et des résultats obtenus ont été effectués sur différents projets métagénomiques. Les performances ont été évaluées sur 29 jeux du projet de séquençage du métro de New-York [124] ($\sim$2.3 millions de lectures Illumina de 300 bp par jeu). METAFAST a requis 82 minutes de temps de traitement et 14 Go de mémoire. Sa distance a été comparée à des distances de Bray-Curtis taxonomiques sur 157 jeux provenant d'une étude de la flore intestinal humaine [121]. METAFAST obtient de bonnes corrélations de Spearman comprises entre 0.8 et 0.86.

Dans la stratégie de METAFAST, l'assemblage est un moyen de sélectionner et de compacter des k-mers afin de réduire leur dimension. Cette stratégie est intéressante car la sélection se base sur des critères qui ont un sens biologique, plutôt qu'aléatoires. La variabilité des comptages de k-mers est également atténuée par le fait de les considérer sur l'ensemble d'une composante plus longue. Cependant, comme nous l'avons vu dans le chapitre introductif, sur des environnements plus complexes, la phase d'assemblage risque d'éliminer une quantité d'information importante. Le coût en ressource informatique augmente également car la mémoire utilisée par le graphe de *de-Bruijn* est linéaire avec le nombre de k-mers distincts insérés. Il en va de même pour le temps de traitement requis pour l'extraction des unitigs. La phase d'identification des composantes connexes lors de l'assemblage croisé est également un procédé coûteux en temps de calcul et difficilement parallélisable. Il est fort probable que des composantes chimériques soient extraites de cette phase d'assemblage grossière à cause des régions conservées entre les génomes. Enfin un dernier inconvénient majeur est l'étape d'assemblage croisé. Le graphe de *de-Bruijn* final et les composantes qui en sont extraites ne vont pas être les mêmes selon le nombre de jeux de données considérés et leur composition. La distance entre deux jeux est donc variable selon l'ensemble des données d'entrée.

2.4 Conclusion

La première approche de métagénomique comparative *de novo* a été effectuée dans le cadre du projet GOS pour comparer le contenu génomique de 44 échantillons océaniques en utilisant le logiciel d'alignement BLAST. Une telle approche compare toutes les lectures contre toutes afin de déterminer le pourcentage de séquences similaires entre chaque paire d'échantillons. Cette méthodologie n'est plus envisageable aujourd'hui compte tenu de l'envergure des projets métagénomiques qui comptent des centaines de jeux de données de centaines de millions de lectures chacun.

COMPAREADS est le premier outil qui passe à l'échelle sur deux grands jeux en comparant les séquences sans le processus d'alignement coûteux. COMPAREADS parvient à traiter deux grands jeux de Tara Oceans en une dizaine d'heures et en n'utilisant que 4 Go de mémoire. COMMET étend la méthodologie de COMPAREADS pour comparer efficacement N jeux ($N > 2$). Mais les changements effectués n'offrent pas un gain considérable sur de gros projets. COMPAREADS et COMMET ont été développés au sein de l'équipe qui a encadré cette thèse et ont été une véritable source d'inspiration. Notamment, puisque COMPAREADS voit un des deux jeux à comparer comme un ensemble de k-mers, notre idée fondamentale a été d'aller encore plus loin et de considérer les deux jeux de cette manière. Le k-mer devient alors l'unité de comparaison des métagénomes.

Cette idée n'est pas anodine : l'usage des k-mers est bien établie en bioinformatique des séquences depuis plusieurs années et des observations de la littérature montrent que les comptages de k-mers sont un bon remplacement aux comptages d'espèces. Trois travaux basés sur cette idée ont justement été publiés en parallèle des développements de cette thèse. La première valide les distances obtenues sur une petite plage de k ($k < 16$). Le développement mis en œuvre dans cette étude ne permet pas d'explorer des k-mers plus grands. La seconde, METAFAST, se base sur l'assemblage et est donc limitée en termes de passage à l'échelle. Enfin, MASH est la méthode la plus performante actuellement. Cependant, cet outil ne fournit pas d'indice de comparaison quantitatif basé sur l'abondance des k-mers.

Avec du recul, nos développements ressemblent à la méthode employée par DSM, un outil publié peu avant le début de cette thèse. Cependant, notre expertise en termes de manipulation des k-mers nous a permis de développer une méthode beaucoup plus efficace. Aujourd'hui, l'outil SIMKA, qui est la contribution principale de cette thèse, est le seul à passer à l'échelle sur de grands projets métagénomiques pouvant contenir des centaines, voire des milliers de grands jeux de données, tout en fournissant un large éventail d'indices de comparaison qualitatifs et quantitatifs. La méthode SIMKA est le sujet du chapitre suivant.

Chapitre 3

Simka : nouvelle méthode de métagénomique comparative *de novo* à grande échelle basée sur des *k*-mers

Ce chapitre présente Simka, la méthode que nous avons développée pendant cette thèse pour comparer efficacement de nombreux jeux de données métagénomiques. Simka rentre dans la catégorie des méthodes de comparaison par spectres de k-mers.

La section 3.1 présente la stratégie de Simka. La section 3.2 décrit une nouvelle méthode pour compter efficacement les k-mers de plusieurs jeux. Basée sur ce compteur de k-mers multi-jeux, la section 3.3 montre comment calculer rapidement de nombreuses distances en une seule passe sur l'ensemble des données. La section 3.4 expose quelques détails d'implémentation. Enfin, la section 3.5 compare Simka avec les autres outils de l'état de l'art en termes de performances. L'évaluation de la qualité des distances proposées par Simka est présentée dans le chapitre 4.

3.1 Stratégie

Soit N jeux de données métagénomiques, dénotés $S_1, S_2, S_i, ...S_N$. Notre objectif est de fournir une matrice de distances D de taille $N \times N$ où $D_{i,j}$ représente la distance entre S_i et S_j. De telles distances sont listées dans le tableau 3.1. Le calcul d'une matrice de distances peut être décomposé en deux étapes :

1. **Comptage des *k*-mers**. Chaque jeu de données est représenté par un ensemble de caractéristiques discriminantes. Dans notre cas, il s'agit de son contenu en k-mers. Plus précisément, une matrice de comptages KC de taille $W \times N$ est calculée, où W est le nombre de k-mers distincts parmi tous les jeux de données. $KC_{i,j}$ représente le nombre de fois qu'un k-mer i est présent dans le jeu de données S_j.
2. **Calcul des distances**. Basée sur l'information des comptages de k-mers, la matrice de distances D est calculée. De nombreuses distances (voir tableau 3.1) peuvent ainsi être dérivées de la matrice KC.

En réalité, Simka ne requiert pas d'avoir à disposition l'intégralité de la matrice KC pour commencer le calcul des distances. Mais pour des raisons de simplicité, nous allons dans un premier temps considérer qu'elle est disponible.

L'étape de comptage des k-mers découpe chaque lecture des jeux de données en k-mers et effectue un comptage global. Cela peut être effectué en comptant les k-mers de chaque jeu de données, puis en fusionnant les spectres de k-mers résultants. Le résultat de ces opérations est

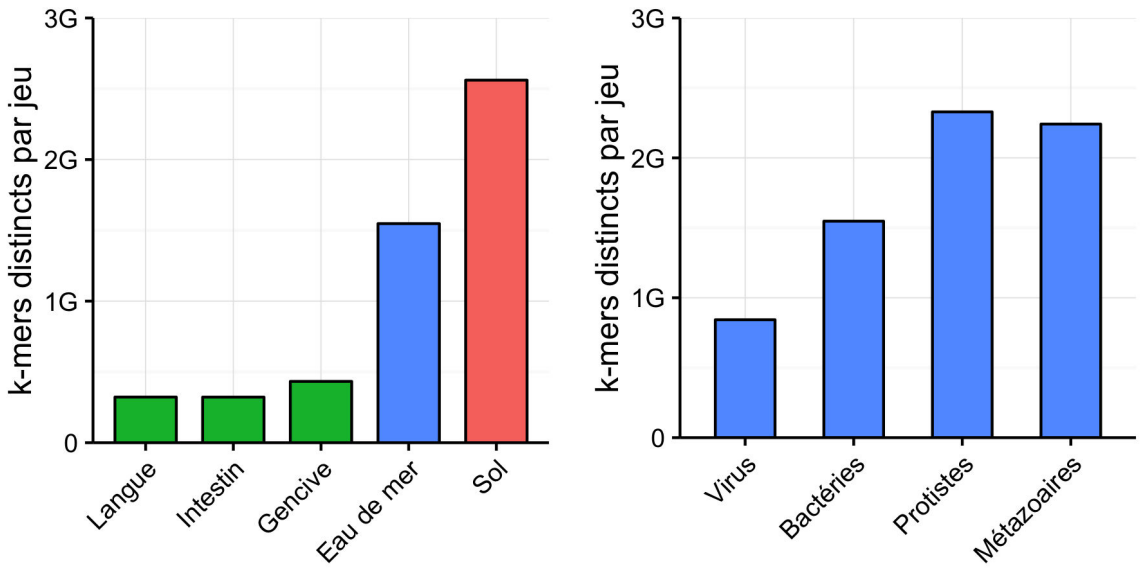

Figure 3.1. Nombre moyen de k-mers distincts par jeu en fonction du type d'environnement et du type d'organismes ($k = 21$). Les statistiques ont été récoltées sur des centaines de jeux métagénomiques du projet HMP (vert), du projet Tara Oceans (bleu) et de projets de sol (rouge, références projets EBI EMG : SRP029969 et SRP008595). La même quantité de données a été considérée dans chaque jeu : 3 milliards de k-mers (∼40 millions de lectures). Les jeux utilisés pour la figure de gauche contiennent majoritairement des bactéries. Les jeux de Tara Oceans considérés dans la figure de droite ont été filtrés par taille d'organismes : de 0 à 0.2 µm (virus), de 0.22 à 3 µm (bactéries), de 0.8 à 5 µm (protistes) et de 5 à 2000 µm (métazoaires).

la matrice KC (de taille $W \times N$). Des algorithmes très performants, tels que KMC2 [113], ont récemment été développés pour extraire le spectre de k-mers d'un jeu de données. Grâce à ces nouveaux outils, l'étape de comptage de chaque jeu peut être effectuée en un temps et un espace mémoire raisonnables, même pour de grands jeux. Cependant, l'inconvénient principal de cette approche vient de l'étape de fusion et de l'énorme quantité d'espace mémoire nécessaire pour stocker la matrice de comptages KC résultante. Cet espace mémoire est calculé de la manière suivante : $Mem_{KC} = W * (8 + 4N)$ octets, où W est le nombre de k-mers distincts, et 8 et 4 sont respectivement le nombre d'octets nécessaires au stockage d'un 31-mer et d'un comptage. Par exemple, le projet Human Microbiome Projet (HMP) [1] entier (690 jeux contenant chacun 45 millions de lectures en moyenne) dénombre 95 milliards de 31-mers distincts et aurait requis un espace de 260 To pour stocker la matrice KC. De plus, le nombre de k-mers distincts peut augmenter drastiquement sur des environnements plus complexes (eau de mer, sol). La figure 3.1 indique que ces milieux génèrent plus de k-mers distincts par jeu. Le type d'organismes étudiés a également un impact considérable car il joue sur la diversité et la taille des génomes présents. Les projets issus de ces milieux complexes généreraient des matrices de comptage encore plus grandes dont le stockage est inconcevable.

Cependant, un regard attentif à la définition des distances (tableau 3.1) montre que, mises à part quelques transformations finales, elles sont toutes additives sur les k-mers distincts. Des contributions indépendantes à la distance peuvent donc être calculées en parallèle à partir d'ensembles disjoints de k-mers, puis agrégées pour construire la distance finale. De plus, chaque contribution peut elle-même être construite de manière itérative, une ligne de la matrice KC à la fois. Dans la suite du manuscrit, on appellera une ligne de la matrice KC un vecteur

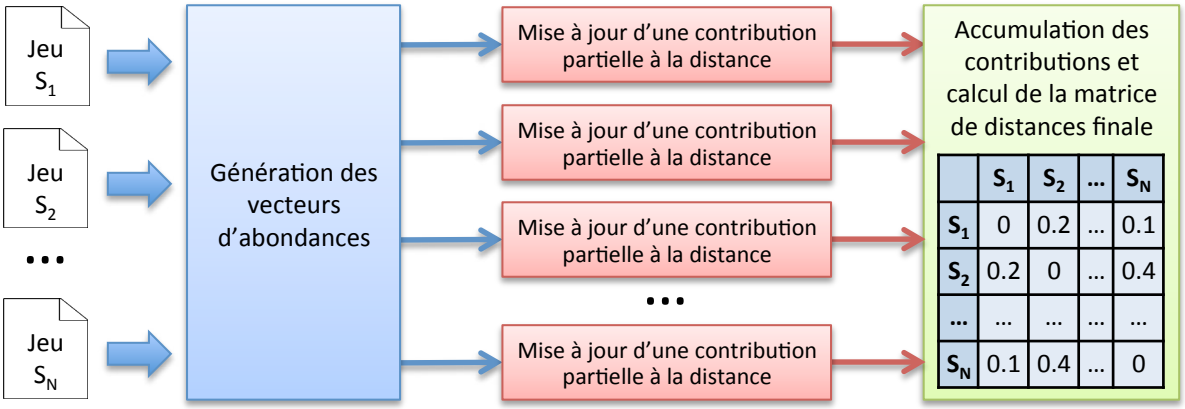

Figure 3.2. Stratégie de SIMKA. La première étape prend en entrée N jeux de données et génère plusieurs flux de vecteurs d'abondances à partir d'ensembles disjoints de k-mers. Un vecteur d'abondances est un k-mer distinct et ses N comptages dans les N jeux de données. Ces vecteurs d'abondances sont pris en entrée de la seconde étape qui met à jour itérativement chaque contribution à la distance en parallèle. Une fois qu'un vecteur d'abondances a contribué à la distance, il n'y a plus besoin de le conserver. L'étape finale cumule chaque contribution et calcule la matrice de distances finale.

d'abondances. Ce dernier représente un k-mer distinct et ses N comptages dans les N jeux de lectures.

Pour récapituler, au lieu de calculer l'intégralité de la matrice de comptages de k-mers KC, le schéma de calcul alternatif que nous proposons consiste à générer successivement les vecteurs d'abondances à partir desquels des contributions indépendantes aux distances peuvent être itérativement calculées en parallèle. L'avantage majeur de cette approche est que l'énorme matrice de comptages KC n'a pas besoin d'être stockée. Cependant, cette approche nécessite le développement d'une nouvelle stratégie pour générer les vecteurs d'abondances. La section suivante décrit la méthode de comptage de k-mers multi-jeux (*Multiset K-mer Counting* - MKC) que nous avons développé dans ce but. Celle-ci est très efficace en temps et en mémoire et peut être parallélisée sur de grandes infrastructures de calcul. Comme illustré dans la figure 3.2, Simka utilise les vecteurs d'abondances générés par le MKC pour calculer les distances du tableau 3.1.

3.2 Comptage de *k*-mers multi-jeux

Partant de N jeux de lectures, l'objectif est de générer les vecteurs d'abondances qui vont nourrir l'étape de calcul des distances. Cette tâche peut être divisée en deux phases :

1. **Comptage des *k*-mers.** Le spectre de k-mers de chaque jeu de données est calculé indépendamment.
2. **Tri-fusion des spectres de *k*-mers.** Un algorithme classique de tri-fusion est appliqué aux N spectres de k-mers pour générer les vecteurs d'abondances. Cet algorithme trie chaque spectre par ordre lexicographique des k-mers. Les N spectres de k-mers triés peuvent ainsi être fusionnés en une passe de lecture.

La limitation majeure de cette approche est l'étape de tri qui peut être très longue sur de grands spectres de k-mers pouvant comporter des milliards d'entrées. En effet, un algorithme de tri a au moins une complexité en temps en $O(n \log n)$ où n est le nombre d'éléments à trier. Pour réduire l'impact de cette étape de tri, nous faisons d'une pierre deux coups en utilisant un algorithme de comptage de k-mers lui-même basé sur le tri des k-mers. Les outils de comptage de

k-mers de référence en bioinformatique (section 2.3.2.1) s'appuient justement sur cette approche. La stratégie présentée précédemment peut alors être redéfinie de la manière suivante :

1. **Tri-comptage.** Le spectre de k-mers de chaque jeu de lectures est calculé indépendamment grâce à un algorithme de comptage de k-mers basé sur le tri de ceux-ci.
2. **Fusion des comptages.** Un algorithme de fusion est appliqué aux N spectres de k-mers (déjà triés) et les vecteurs d'abondances générés.

3.2.1 Tri-comptage

Tous les k-mers d'un jeu de données sont extraits, puis triés par ordre lexicographique. Les k-mers distincts peuvent alors être facilement identifiés et leur nombre d'occurrences calculé. Puisque le nombre de k-mers distincts est généralement très important, l'étape de tri est divisée en deux sous-étapes et fonctionne de la manière suivante : les k-mers sont d'abord séparés en P partitions, chacune stockée sur le disque. Après cette étape préliminaire, chaque partition est triée et comptée indépendamment, puis stockée sur le disque une nouvelle fois. Conceptuellement, à la fin du processus de tri-comptage, nous disposons de $N \times P$ partitions triées. La figure 3.3-A illustre la phase de tri-comptage.

Pour effectuer efficacement cette étape de comptage de k-mers basée sur le tri, nous utilisons des approches de référence, tels que KMC2 [113] et DSK [112], dont les méthodologies ont été détaillées en section 2.3.2.1. Ces outils fournissent notamment un niveau de parallélisation à grain-fin et permettent l'exploitation des capacités des processeurs multi-cœurs actuels. Un second niveau de parallélisation gros-grain est obtenu par le comptage indépendant de chaque jeu de données. N processus peuvent donc être exécutés en parallèle, chacun traitant un jeu de données spécifique. Le processus global de tri-comptage convient donc particulièrement bien à de grandes infrastructures de calcul possédant des centaines de nœuds, et où chaque nœud implémente des systèmes à 8 ou 16 cœurs. Afin d'effectuer l'étape de fusion efficacement (détaillée dans la section suivante), la même fonction de répartition des k-mers doit être utilisée pour tous les jeux de données (même nombre de partitions et même fonction de hachage). Ainsi, en pratique, nous utilisons l'algorithme de comptage DSK qui a été développé au sein de notre équipe, et dans lequel nous avons donc pu facilement implémenter ce changement.

Afin de limiter les accès disque, les partitions sont compressées. Puisque chaque partition est triée, chaque k-mer peut avoir un grand préfixe commun avec le k-mer précédent. Dans notre méthodologie, cette redondance est capturée grâce à une approche de compression par dictionnaire. Cette approche a été implémentée dans de nombreuses librairies et outils de compression très connus, tels que ZLIB [125].

3.2.2 Fusion des comptages

Pour des raisons pratiques, les logiciels de comptage de k-mers fournissent généralement un unique fichier de comptage de k-mers en sortie et perdent l'information du partitionnement des k-mers. Ici, nous montrons que ce partitionnement peut être utilisé avantageusement pour générer les vecteurs d'abondances en parallèle.

Les N fichiers associés à une partition P_i sont pris en entrée d'un processus de fusion. Pour rappel, ces fichiers contiennent chacun un sous-spectre de k-mers trié. Un algorithme de tri-fusion peut donc être appliqué pour générer directement les vecteurs d'abondances.

Jusqu'à P processus de fusion peuvent être exécutés indépendamment, résultant chacun dans la génération de P vecteurs d'abondances en parallèle et permettant donc le calcul de P contributions à la distance simultanément. Il est important de noter que les vecteurs d'abondances n'ont pas besoin d'être stockés sur le disque. Ils sont seulement utilisés comme un flux d'entrée

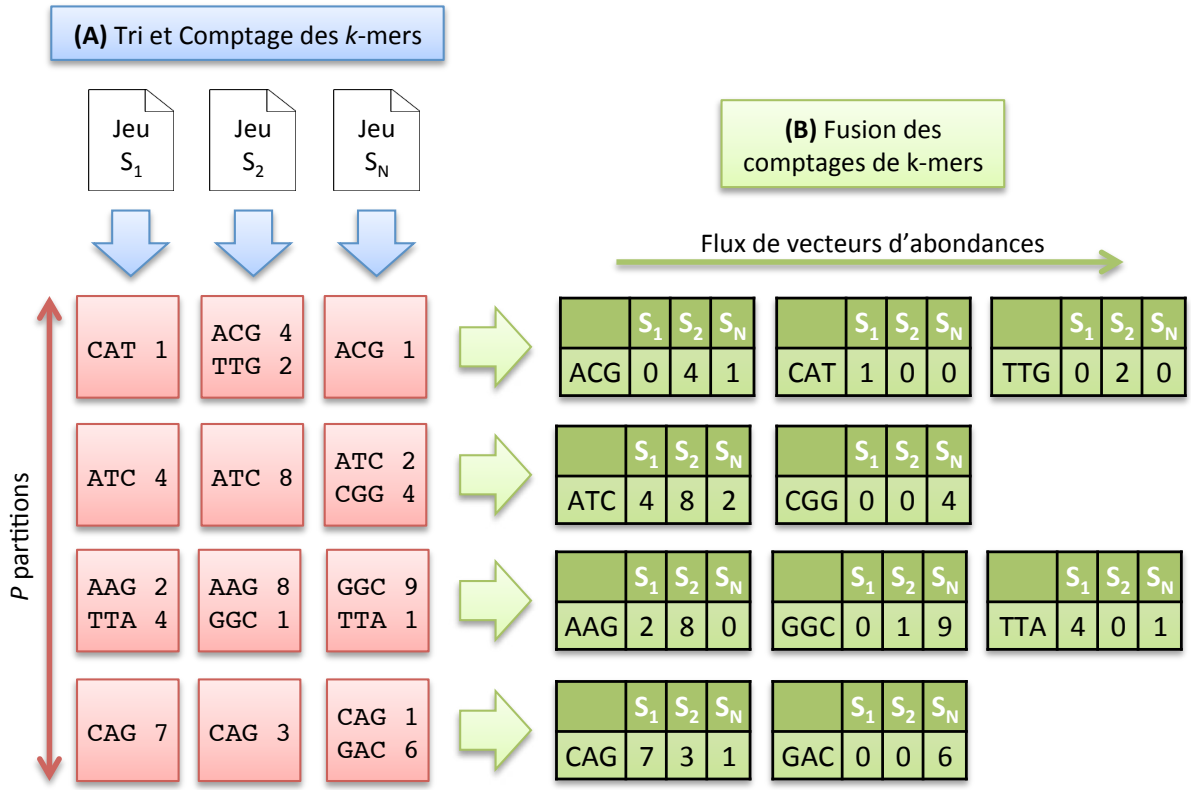

Figure 3.3. Stratégie de comptage de *k*-mers multi-jeux avec k=3. (**A**) Un processus de comptage, représenté par une flèche bleue, compte les k-mers d'un jeu de données indépendamment des autres. Chaque processus produit une colonne de P partitions (carrés rouges) contenant des k-mers triés et leurs comptages. (**B**) Un processus de fusion, représenté par une flèche verte, fusionne une ligne de N partitions. Il produit des vecteurs d'abondances, représentés en vert, qui nourrissent l'étape de calcul des distances.

pour l'étape suivante de calcul des distances. La figure 3.3-B illustre la phase de fusion des comptages de k-mers.

3.2.3 Filtre d'abondance des *k*-mers

Les erreurs de séquençage génèrent des k-mers erronés de faible abondance. Une simple erreur peut générer jusqu'à k k-mers distincts erronés. Filtrer ces k-mers améliore considérablement les performances de Simka puisque cela réduit le nombre total de k-mers distincts. Cependant, en métagénomique, les k-mers rares peuvent également provenir d'espèces rares. L'impact de ce filtre sur la qualité des distances est évalué et discuté dans la section 4.1.

Ce filtre peut être activé pendant la phase de comptage des k-mers. Pour un jeu donné, seuls ses k-mers ayant une abondance égale ou plus grande qu'un seuil d'abondance donné sont conservés. Par défaut, ce seuil est fixé à 2. Les k-mers qui passent ce filtre sont appelés des k-mers solides.

3.3 Calcul des distances

De nombreux indices de comparaison ont été définis par les écologues pour comparer les communautés. Ils se regroupent en deux familles principales (voir [86] pour une classification plus fine) : les indices qualitatifs et les indices quantitatifs. La première famille traite les éléments de

manière égale, qu'ils soient rares ou très abondants. À l'inverse, les indices quantitatifs s'appuient sur les variations d'abondance des éléments. Ici, les éléments abondants ont plus de poids que les éléments rares. Des études montrent que les différentes distances peuvent capturer différentes caractéristiques des jeux de données [86, 126]. D'autres conseillent d'analyser les données sur la base de plusieurs distances [127]. Pour ces raisons, nous avons fait le choix de calculer une grande partie des distances existantes. Nous avons étudié leur formulation afin de développer un algorithme pour les calculer efficacement en une passe sur les données.

Pour des raisons de simplicité, le calcul de la distance quantitative de Bray-Curtis est tout d'abord expliqué. Toutes les autres distances présentées ensuite peuvent être calculées en suivant le même schéma, avec seulement quelques adaptations mineures.

3.3.1 Calcul de la distance de Bray-Curtis

La distance de Bray-Curtis est donnée par l'équation suivante :

$$BC(S_i, S_j) = 1 - 2\frac{\sum_{w \in S_i \cup S_j} min(N_{S_i}(w), N_{S_j}(w))}{\sum_{w \in S_i} N_{S_i}(w) + \sum_{w \in S_j} N_{S_j}(w)} \tag{3.1}$$

où w est un k-mer et $N_{S_i}(w)$ est l'abondance de w dans le jeu de lectures S_i.

Cette équation implique des termes marginaux (ou spécifiques à un jeu de lectures) et des termes croisés. Par exemple, le terme marginal $\sum_{w \in S_i} N_{S_i}(w)$ est le nombre total de k-mers dans le jeu de lectures S_i. Il va agir comme une constante de normalisation. Les termes croisés capturent la (dis)similarité entre les deux jeux de données. Par exemple, le terme croisé $\sum_{w \in S_i \cup S_j} \min(N_{S_i}(w), N_{S_j}(w))$ est le nombre total de k-mers dans l'intersection des jeux de données S_i et S_j. Les termes marginaux et croisés sont ensuite combinés pour calculer la distance finale.

L'algorithme 1 montre qu'il est simple de calculer la matrice de distances entre les N jeux de lectures à partir des vecteurs d'abondances. L'entrée de cet algorithme est fournie par le compteur de k-mers multi-jeux (MKC). Cette entrée consiste en P flux de vecteurs d'abondances et les termes marginaux de la distance, c'est à dire le nombre de k-mers dans chaque jeu de lectures, déterminé pendant la première phase du MKC qui compte les k-mers.

Une matrice, désignée $M_\cap$, de dimension $N \times N$ est initialisée (ligne 1) pour enregistrer la valeur finale du terme croisé de chaque paire de jeux de lectures. P processus indépendants sont lancés (ligne 2) pour calculer P matrices de termes croisés partielles, dénotées $M_{\cap part}$ (ligne 3), en parallèle. Chaque processus parcourt son flux de vecteurs d'abondances (ligne 4). Pour chaque vecteur d'abondances, on boucle sur chaque paire de jeux de lectures possibles (lignes 5-6). La matrice $M_{\cap part}$ est mise à jour (ligne 8) si le k-mer est partagé, signifiant qu'il a une abondance positive dans les deux jeux de lectures S_i et S_j (ligne 7). Puisque la matrice de distances est symétrique avec une diagonale nulle, les calculs se limitent au triangle supérieur de la matrice $M_{\cap part}$. Le vecteur d'abondances courant est ensuite libéré. Chaque processus écrit sa matrice $M_{\cap part}$ sur le disque quand son flux est terminé (ligne 9).

Quand tous les flux ont été traités, l'algorithme lit chaque matrice $M_{\cap part}$ et l'accumule à la matrice de termes croisés complète $M_\cap$ (lignes 10-11). La dernière boucle (lignes 13 à 16) calcule la distance de Bray-Curtis pour chaque paire de jeux de lectures et remplit la matrice de distances qui est la sortie de Simka.

Le nombre de vecteurs d'abondances émis par le MKC est égal à W_s, le nombre total de k-mers distincts solides dans les N jeux de lectures. Cet algorithme a donc une complexité en temps de $O(W_s \times N^2)$.

Algorithme 1 : Calcul de la distance de Bray-Curtis (équation 3.1) entre N jeux de lectures.

Entrées :
- V_s : vecteur de taille P représentant les flux de vecteurs d'abondances
- $V_\cup$: vecteur de taille N contenant le nombre de k-mers dans chaque jeu de lectures

Output : une matrice de distances $Dist$

1 $M_\cap \leftarrow$ matrice carrée de taille $N \times N$ // nombre de k-mers dans l'intersection de chaque paire de jeux de lectures
2 **En parallèle: pour chaque** *flux de vecteurs d'abondances* S ***dans*** V_s **faire**
3 $M_{\cap part} \leftarrow$ matrice carrée de taille $N \times N$ // partie de $M_\cap$
4 **pour chaque** *vecteurs d'abondances* v ***dans*** S **faire**
5 **pour** $i \leftarrow 0$ **à** $N-1$ **faire**
6 **pour** $j \leftarrow i+1$ **à** $N-1$ **faire**
7 **si** $v[i] > 0$ *et* $v[j] > 0$ **alors**
8 $M_{\cap part}[i,j] \leftarrow M_{\cap part}[i,j] + \min(v[i], v[j])$
9 Écrire $M_{\cap part}$ sur le disque
10 **pour chaque** *matrice écrite* $M_{\cap part}$ **faire**
11 $M_\cap \leftarrow M_\cap + M_{\cap part}$
12 $Dist \leftarrow$ matrice carrée de taille $N \times N$ // matrice de distances finales
13 **pour** $i \leftarrow 0$ **à** $N-1$ **faire**
14 **pour** $j \leftarrow i+1$ **à** $N-1$ **faire**
15 $Dist[i,j] = 1 - 2 * M_\cap[i,j] \ / \ (V_\cup[i] + V_\cup[j])$
16 $Dist[j,i] = 1 - 2 * M_\cap[i,j] \ / \ (V_\cup[i] + V_\cup[j])$
17 **retourner** $Dist$

3.3.2 Autres distances

La plupart des distances écologiques, incluant celles mentionnées dans [86], peuvent être exprimées pour une paire de jeux de lectures S_i et S_j comme :

$$Distance(S_i, S_j) = g\left(\sum_{w \in S_i \cup S_j} f\left(N_{S_i}(w), N_{S_j}(w), C_{S_i}, C_{S_j}\right)\right) \tag{3.2}$$

où g et f sont des fonctions simples, et C_{S_i} est un terme marginal (spécifique) du jeu de lectures S_i, généralement un scalaire. Dans la plupart des distances, C_{S_i} est simplement le nombre total de k-mers dans S_i. En revanche, la valeur de f correspond aux termes croisés et requiert la connaissance de $N_{S_i}(w)$ et $N_{S_j}(w)$ (et potentiellement C_{S_i} et C_{S_j} également). Par exemple, pour le calcul de la distance de Bray-Curtis (équation 3.1), nous avons $C_{S_i} = \sum_{w \in S_i} N_{S_i}(w)$, $g(x) = 1 - 2x$ et $f(x, y, X, Y) = \min(x, y)/(X + Y)$.

Ces distances peuvent être calculées en une seule passe sur les données en utilisant des versions légèrement différentes de l'algorithme 1. Les termes marginaux C_{S_i} sont déterminés pendant la première étape du MKC qui compte les k-mers de chaque jeu de lectures. Les termes croisés impliquant f sont calculés et sommés lignes 7-8 (mais les instructions exactes dépendent de la nature de f). Finalement, les distances finales sont calculées en lignes 15-16 et dépendent de f et g.

Les distances qualitatives forment une catégorie spéciale des distances écologiques : elles

peuvent toutes être exprimées en termes de quantités a, b et c où a est le nombre de k-mers distincts partagés par S_i et S_j, et b et c sont respectivement le nombre de k-mers distincts présents seulement dans S_i et S_j. Ces distances s'insèrent parfaitement bien dans le cadre présenté précédemment avec $a = \sum_{w \in S_i \cap S_j} 1_{\{N_{S_i}(w) N_{S_j}(w) > 0\}}$, $C_{S_i} = \sum_{w \in S_i} 1_{\{N_{S_i}(w) > 0\}} = a + b$ et de la même manière $C_{S_j} = a + c$. Ainsi, a est un terme croisé et les termes b et c peuvent être déduits de a et des termes marginaux.

De la même manière, Chao *et al.* [128] introduisent une variation des distances qualitatives en leur incorporant des informations d'abondances. L'idée principale est de remplacer les quantités "rude" telles que $a/(a+b)$, la fraction de k-mers distincts de S_i partagés avec S_j, par des quantités probabilistes "douces" : ici, la probabilité $U \in [0, 1]$ qu'un k-mer de S_i soit aussi trouvé dans S_j. De même, la fraction "rude" $a/(a+c)$ de k-mers distincts de S_j partagés avec S_i est remplacée par une probabilité "douce" V qu'un k-mer de S_j soit aussi trouvé dans S_i. U et V jouent le même rôle que a, b et c dans les distances qualitatives et sont suffisantes pour calculer des variantes appelées AB-Jaccard, AB-Ochiai et AB-Sorensen. Cependant, contrairement aux quantités a, b et c qui peuvent être observées à partir des données, U et V ne sont pas connus en pratique et doivent être estimés à partir des données. Nous avons mis en œuvre la plus simple des estimations présentées dans [128], qui se prête parfaitement bien à la nature additive et distribuée de SIMKA : $U = Y_{S_i S_j}/C_{S_i}$ et $V = Y_{S_j S_i}/C_{S_j}$ où $Y_{S_i S_j} = \sum_{w \in S_i \cap S_j} N_{S_i}(w) 1_{\{N_{S_j}(w) > 0\}}$ et $C_{S_i} = \sum_{w \in S_i} N_{S_i}(w)$. Notons que $Y_{S_i S_j}$ correspond aux termes croisés et est asymétrique, i.e. $Y_{S_i S_j} \neq Y_{S_j S_i}$. Intuitivement, U est la fraction de k-mers de S_i également trouvés dans S_j et donne bien plus de poids aux k-mers abondants que sa version qualitative $a/(a+b)$.

Le tableau 3.1 montre la collection de distances calculées par SIMKA en remplaçant les comptages d'espèces par des comptages de k-mers. Cette collection inclut des distances quantitatives, qualitatives et des variantes des distances qualitatives prenant en compte l'abondance des k-mers. Le tableau fournit également leur expression en termes de C_i, f et g, en adoptant la notation de l'équation 3.2.

La nature additive de ces distances calculées sur les k-mers est nécessaire pour atteindre une complexité linéaire sur le nombre de k-mers distincts solides W_s et pour pouvoir paralléliser les calculs. Ainsi, notre algorithme n'est pas adaptable à d'autres distances plus complexes qui prennent par exemple en compte la similarité entre les espèces [126], ou qui requièrent le calcul de distances d'édition entre les k-mers et nécessitent donc une comparaison des k-mers tous contre tous.

3.3.3 Distances simples et distances complexes

D'un point de vue algorithmique, une optimisation majeure peut être apportée à l'algorithme 1. Cela concerne le calcul du terme croisé des distances (ligne 7 et 8). Dans le cas de la distance de Bray-Curtis, ce calcul n'a lieu que si une paire de jeux partage le k-mer courant (ligne 7). En effet, si un jeu ne possède pas le k-mer courant, son terme croisé vaudra toujours 0. Cette ligne agit donc comme un filtre pour ne pas appeler inutilement le calcul des termes croisés qui peut être coûteux en temps selon la définition des distances. Cependant, lorsque le nombre N de jeux est grand ($N > 100$), ce filtre a lui même un impact considérable sur le temps de traitement puisqu'il est appelé un nombre quadratique de fois sur N pour chaque k-mer distinct.

Il est possible de ne plus employer explicitement ce filtre tout en conservant son usage. Cette optimisation se base sur la nature très creuse des vecteurs d'abondances. C'est à dire que les jeux métagénomiques partagent peu de k-mers globalement. Par exemple, dans le cas du projet Tara Oceans, 88% des k-mers distincts sont spécifiques à un jeu et n'ont même pas besoin d'être pris en compte par l'algorithme 1 (tests personnels sur 150 grands jeux de lectures). En effet, ces k-mers spécifiques n'interviennent que dans le calcul des termes marginaux dont la valeur

Nom	Définition	C_{S_i}	$f(x, y, X, Y)$	$g(x)$
	Distances quantitatives			
Chord	$\sqrt{2 - 2\sum_w \frac{N_{S_i}(w)N_{S_j}(w)}{C_{S_i}C_{S_j}}}$	$\sqrt{\sum_w N_{S_i}(w)^2}$	$\frac{xy}{XY}$	$\sqrt{2-2x}$
Hellinger	$\sqrt{2 - 2\sum_w \frac{\sqrt{N_{S_i}(w)N_{S_j}(w)}}{\sqrt{C_{S_i}C_{S_j}}}}$	$\sum_w N_{S_i}(w)$	$\frac{\sqrt{xy}}{\sqrt{XY}}$	$\sqrt{2-2x}$
Whittaker	$\frac{1}{2}\sum_w \frac{\lvert N_{S_i}(w)C_{S_j} - N_{S_j}(w)C_{S_i}\rvert}{C_{S_i}C_{S_j}}$	$\sum_w N_{S_i}(w)$	$\frac{\lvert xY - yX\rvert}{XY}$	$\frac{x}{2}$
Bray-Curtis	$1 - 2\sum_w \frac{\min(N_{S_i}(w), N_{S_j}(w))}{C_{S_i} + C_{S_j}}$	$\sum_w N_{S_i}(w)$	$\frac{\min(x,y)}{X+Y}$	$1-2x$
Kulczynski	$1 - \frac{1}{2}\sum_w \frac{(C_{S_i} + C_{S_j})\min(N_{S_i}(w), N_{S_j}(w))}{C_{S_i}C_{S_j}}$	$\sum_w N_{S_i}(w)$	$\frac{(X+Y)\min(x,y)}{XY}$	$1-\frac{x}{2}$
Jensen-Shannon	$\sqrt{\frac{1}{2}\sum_w \left[\frac{N_{S_i}(w)}{C_{S_i}} \log \frac{2C_{S_j}N_{S_i}(w)}{C_{S_j}N_{S_i}(w) + C_{S_i}N_{S_j}(w)} + \frac{N_{S_j}(w)}{C_{S_j}} \log \frac{2C_{S_i}N_{S_j}(w)}{C_{S_j}N_{S_i}(w) + C_{S_i}N_{S_j}(w)}\right]}$	$\sum_w N_{S_i}(w)$	$\frac{x}{X}\log\frac{2xY}{xY+yX} + \frac{y}{Y}\log\frac{2yX}{xY+yX}$	$\sqrt{\frac{x}{2}}$
Canberra	$\frac{1}{a+b+c}\sum_w \left\lvert \frac{N_{S_i}(w) - N_{S_j}(w)}{N_{S_i}(w) + N_{S_j}(w)} \right\rvert$	–	$\left\lvert \frac{x-y}{x+y} \right\rvert$	$\frac{1}{a+b+c}x$
	Distances qualitatives			
Chord/Hellinger	$\sqrt{2\left(1 - \frac{a}{\sqrt{(a+b)(a+c)}}\right)}$	–	–	–
Whittaker	$\frac{1}{2}\left(\frac{b}{a+b} + \frac{c}{a+c} + \left\lvert \frac{a}{a+b} - \frac{a}{a+c} \right\rvert\right)$	–	–	–
Bray-Curtis/Sorensen	$\frac{b+c}{2a+b+c}$	–	–	–
Kulczynski	$1 - \frac{1}{2}\left(\frac{a}{a+b} + \frac{a}{a+c}\right)$	–	–	–
Ochiai	$1 - \frac{a}{\sqrt{(a+b)(a+c)}}$	–	–	–
Jaccard	$\frac{b+c}{a+b+c}$	–	–	–
	Variantes quantitatives (AB) de distances qualitatives			
AB-Jaccard	$1 - \frac{UV}{U+V-UV}$	–	–	–
AB-Ochiai	$1 - \sqrt{UV}$	–	–	–
AB-Sorensen	$1 - \frac{2UV}{U+V}$	–	–	–

Table 3.1. Définition des distances calculées par Simka. Toutes les distances quantitatives peuvent être exprimées en termes de C_S, $f = f(x, y, X, Y)$ et $g = g(x)$, en utilisant la notation de l'équation 3.2, et calculées en une passe sur les données. Les distances qualitatives (respectivement leur variante prenant en compte l'abondance) peuvent aussi être calculées en une passe en calculant a, b et c (respectivement U et V).

est déjà déterminée suite à la phase de comptage des k-mers. Parmi les 12% de k-mers distincts partagés, 75% sont partagés par seulement deux jeux de données, 15% par 3 jeux, etc. Pour profiter avantageusement de cette situation, les vecteurs d'abondances sont transformés en une version parcimonieuse : les zéros ne sont plus stockés et tout comptage supérieur à 0 est remplacé par une paire (p, c), où p est la position du comptage c dans le vecteur d'abondances. Il n'y a alors plus besoin de la ligne 7 de l'algorithme 1 puisque les jeux ne possédant pas le k-mer courant ont déjà été filtrés.

Cette optimisation permet un gain de temps de calcul des distances conséquent. Cette astuce algorithmique permet donc de briser la nature quadratique du calcul des distances pour une grande majorité des k-mers distincts. Cependant, elle n'est pas directement applicable à toutes les distances écologiques. C'est ainsi que, d'un point de vue algorithmique, nous avons défini deux catégories de distances : les distances simples et les distances complexes. Les distances simples, incluant Bray-Curtis, ont besoin d'être mises à jour seulement pour chaque paire (S_i, S_j) telle que $N_{S_i} > 0$ et $N_{S_j} > 0$ tandis que les distances complexes ont besoin d'être mises à jour pour chaque paire telle que $N_{S_i} > 0$ ou $N_{S_j} > 0$. Les distances complexes requièrent par conséquent beaucoup plus d'opérations. Si N est la taille originale des vecteurs d'abondances et n est le nombre de jeux S_i tel que $N_{S_i} > 0$ ($n << N$), l'algorithme 1 engendre N^2 opérations par k-mer distinct, les distances simples en nécessitent n^2 et les distances complexes en requièrent $n * N$. Les distances complexes profitent donc tout de même en partie de cette optimisation en ne traitant plus les paires qui ne possèdent pas le k-mer courant.

La distinction de ces deux types de distances nous a invité à réviser leur formulation. Par exemple, dans la littérature, deux définitions de la distances de Bray-Curtis existent :

$$BC(S_i, S_j) = 1 - 2\frac{\sum_{w \in S_i \cup S_j} min(N_{S_i}(w), N_{S_j}(w))}{\sum_{w \in S_i} N_{S_i}(w) + \sum_{w \in S_j} N_{S_j}(w)} = \frac{\sum_{w \in S_i \cup S_j} |(N_{S_i}(w) - N_{S_j}(w))|}{\sum_{w \in S_i} N_{S_i}(w) + \sum_{w \in S_j} N_{S_j}(w)}$$

Les deux aboutissent au même résultat mais la première est formulée comme une distance simple alors que la seconde est une distance complexe. Il est important de noter que parmi les distances présentées dans le tableau 3.1, toutes les distances sont simples, à l'exception des distances de Whittaker, Jensen-Shannon et Canberra.

3.4 Implémentation

Cette section fournit quelques détails d'implémentation du programme Simka et nos choix quant à la représentation des spectres de k-mers sur le disque.

Représentation des spectres de *k*-mers dans SIMKA. La première étape du MKC (section 3.2.1) compte les k-mers d'un jeu et stocke le spectre résultant sur le disque. Lors de la seconde étape (section 3.2.2), les spectres sont lus en parallèle et fusionnés. Cependant, plus le nombre de jeux N est grand, plus le nombre de fichiers ouverts augmente et plus long devient leur temps de lecture. Cela provient de limitations technologiques du système de gestion de fichiers. Notamment, lorsque $N > 4000$, il devient quasiment impossible de lire les spectres ou d'en ouvrir de nouveaux.

Pour parvenir à comparer de nombreux spectres sans impact du système de gestion de fichiers, des fusions intermédiaires de N' spectres ($N' < N$) en un seul sont effectués. La figure 3.4 (gauche) illustre la représentation classique d'un spectre. Chaque fichier contient simplement des k-mers distincts triés et leur comptage. Par comparaison, la figure 3.4 (droite) donne le format de stockage de plusieurs spectres en un seul fichier. Chaque spectre est associé à un identifiant

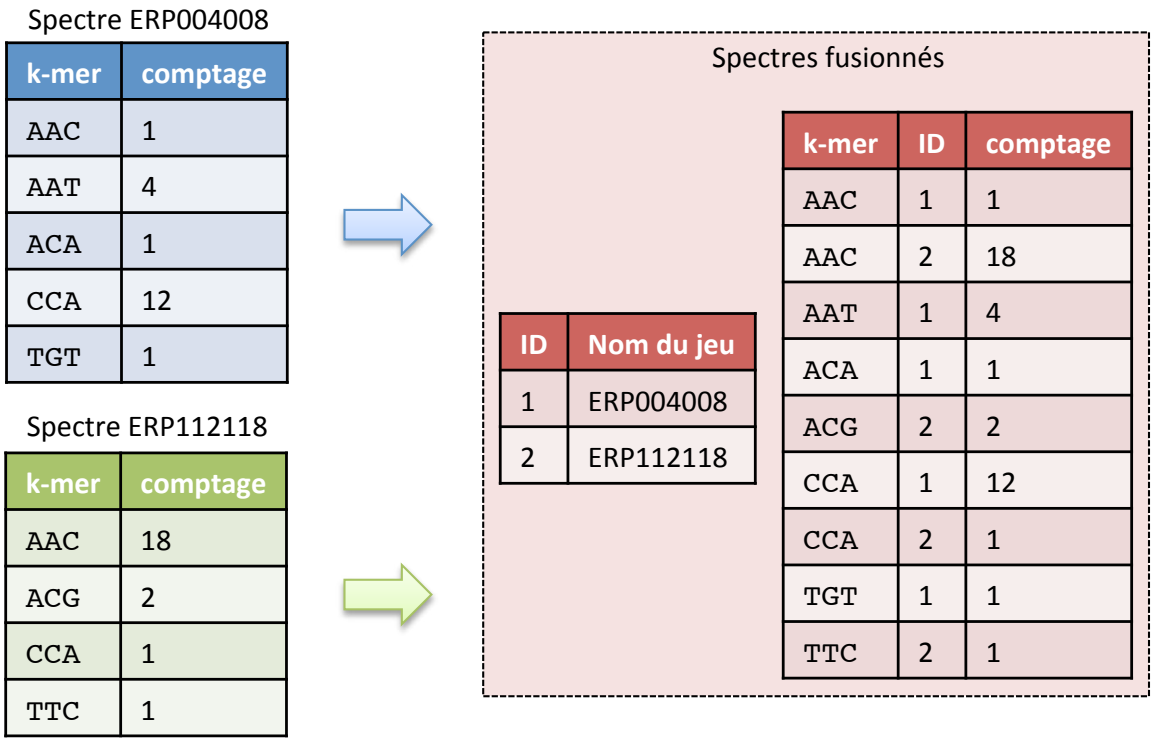

Figure 3.4. Représentation des spectres de k-mers dans Simka. La représentation classique d'un spectre est donnée à gauche pour deux spectres provenant de deux jeux nommés ERP004008 et ERP112118. Le spectre rouge représente la fusion de ces deux spectres en un seul fichier. Un identifiant unique est associé à chaque jeu. Une première table stocke le nom des jeux associé à leur identifiant. Une seconde table contient les k-mers distincts des deux spectres par ordre lexicographique, l'identifiant de leur jeu et leur comptage.

unique. Le nom des jeux, qui peut être long, est stocké une seule fois dans une table à part et associé à son identifiant. Une seconde table contient les deux spectres fusionnés. Chaque k-mer distinct est associé à son comptage et à l'identifiant de son jeu. Ce format est parcimonieux. Il ne stocke pas les zéros contrairement à la matrice de comptages KC décrite en section 3.1 (qui atteint des centaines de To dans les exemples donnés). L'idée derrière ce format part du principe que cette matrice est très creuse, ce qui est le cas en pratique. Comme indiqué dans la section suivante, cette représentation permet le stockage de la matrice KC sur le disque en utilisant un espace de l'ordre de la taille des jeux de données d'entrée.

Suite de programmes de SIMKA. Simka est basé sur l'API GATB (Genome Analysis Tool Box) [129], une bibliothèque C++ optimisée pour manipuler de grands ensembles de k-mers. C'est un logiciel *open source*, distribué sous la licence GNU affero GPL, disponible en téléchargement sur le site web de GATB : `https://gatb.inria.fr/software/simka/`.

Une suite de programmes est mise à disposition pour effectuer tous les calculs nécessaires et permettre leur parallélisation sur de grandes infrastructures de calcul :

— **simka-count.** Ce programme transforme un jeu de lectures en son spectre de k-mers découpé en P partitions. Cette partie de Simka est basée sur l'outil de comptage de k-mers DSK, qui implémente la méthodologie décrite dans KMC2 et dont le code est intégré et modifiable dans GATB. DSK a notamment été modifié pour conserver l'information du partitionnement des k-mers afin de paralléliser l'étape de génération des vecteurs

d'abondances et de calcul des distances. La représentation d'un spectre de k-mers d'un unique jeu de lectures est illustrée par la figure 3.4 (gauche).

— **simka-merge.** Lorsque le nombre de spectres de k-mers à traiter est grand ($N > 200$ par défaut), ce programme effectue des fusions intermédiaires de N' spectres ($N' < N$) en un seul. La représentation d'un spectre contenant les comptages de plusieurs jeux est montrée par la figure 3.4 (droite).

— **simka-distance.** Génère les vecteurs d'abondances et calcule la contribution aux distances écologiques pour une partition P_i des N spectres de k-mers.

— **simka-distanceFinal.** Agglomère les P contributions aux distances, applique les transformations finales et fournit les matrices de distances.

Le programme principal "simka" enrobe cette suite de programmes et gère le lancement, la parallélisation et la synchronisation des processus. Simka possède un système de points de contrôle qui permet de ne pas avoir à tout recalculer si l'exécution s'est arrêtée en cours de traitement ou si l'utilisateur souhaite ajouter de nouveaux jeux de données. Pour déployer Simka sur un cluster de calcul, il suffit de lui fournir la commande de soumission des jobs du cluster en question.

3.5 Évaluation des performances

Les performances de Simka sont évaluées en termes de temps d'exécution, d'empreinte mémoire et d'utilisation du disque, et comparées à celles des outils de l'état de l'art. L'évaluation de la qualité des distances de Simka est abordée dans le chapitre 4.

Les expérimentations ont été conduites sur les données du Human Microbiome Project (HMP) [1] qui est actuellement un des plus gros projets métagénomiques publiques en termes de jeux de lectures plein-génome : 690 échantillons prélevés de différents tissus (`http://www.hmpdacc.org/HMASM/`). Le projet entier a été séquencé via la technologie Illumina et contient en tout 2×16 milliards de lectures appairées d'une centaine de nucléotides réparties non uniformément à travers les 690 jeux de données.

3.5.1 Performances sur de petits jeux de lectures

Le passage à l'échelle de Simka a tout d'abord été évalué sur de petits jeux de données du projet HMP. Le nombre de jeux à traiter augmente de 2 à 40. Lorsqu'une distance simple est calculée, comme la distance de Bray-Curtis, Simka montre un temps d'exécution ayant une allure linéaire avec le nombre de jeux comparés (figure 3.5-A). Comme espéré, le comptage des k-mers de chaque jeu de données (MKC-Comptage) consomme la majorité du temps, notamment à cause d'une forte utilisation du disque. Le temps de génération des vecteurs d'abondances (MKC-Fusion) est extrêmement rapide. Le calcul des distances simples est la partie la plus rapide grâce aux optimisations (présentées en section 3.3.3) pour briser sa complexité normalement quadratique sur le nombre de jeux de lectures. La seule exception au comportement linéaire global est le calcul des distances complexes dont l'allure augmente quadratiquement avec N.

Lors des comparaisons aux autres outils de l'état de l'art, à savoir Commet, Metafast et Mash, Simka a été paramétré afin de ne calculer que la distance de Bray-Curtis, puisque les autres outils ne calculent également qu'une seule distance simple. DSM n'a pas été testé car il nécessite une infrastructure de calcul spécifique pour distribuer la mémoire sur plusieurs machines. Les figures 3.5-B-C-D montrent respectivement, le temps CPU, l'empreinte mémoire et l'utilisation du disque maximum de chaque outil par rapport à un nombre croissant de jeux de lectures N. Mash est définitivement l'outil qui passe le mieux à l'échelle mais les limitations de la distance qu'il calcule sont montrées dans la section suivante. Commet est le seul outil à

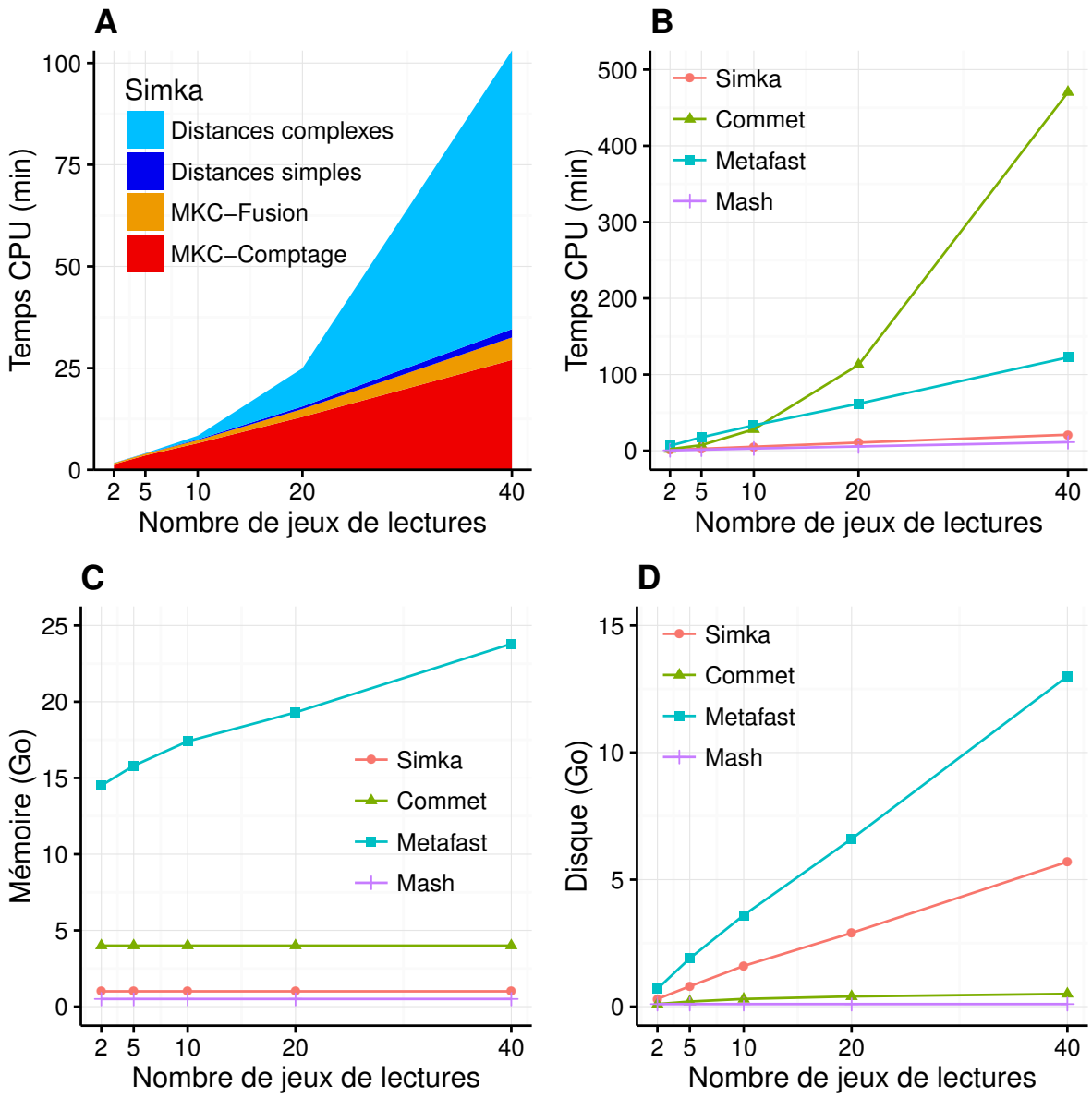

Figure 3.5. Performances de SIMKA et des outils de l'état de l'art par rapport à un nombre N de jeux de lectures à comparer. Chaque jeu de données est composé de deux millions de lectures. Tous les outils ont été lancés sur une machine équipée d'un processeur Intel E5-2640 de 20 cœurs (2.50 GHz) et 264 Go de mémoire. (**A**) et (**B**) Temps CPU par rapport à N. Pour (**A**), les couleurs correspondent aux différentes étapes principales de Simka. (**C**) Empreinte mémoire par rapport à N. (**D**) Utilisation du disque par rapport à N. Les paramètres et lignes de commande utilisés pour chaque outil sont détaillés en annexe (tableau 1).

montrer un comportement quadratique avec N. Pour $N = 40$, Simka est 6 fois plus rapide que Metafast et 22 fois plus rapide que Commet. Tous les outils, à l'exception de Metafast, ont une empreinte mémoire maximale constante par rapport à N. L'utilisation du disque des quatre outils augmente de manière linéaire avec N. Le coefficient linaire est plus grand pour Simka et Metafast, mais cela reste raisonnable pour Simka, puisque son utilisation totale du disque est d'environ la moitié de la taille des données d'entrée qui était de 11 Go pour $N = 40$.

En résumé, Simka et Mash semblent être les seuls outils capables de traiter de grands jeux

HMP - 690 jeux de lectures - 3727 Go - 2×16 milliards de lectures appairées		
	Sans filtre	**Avec filtre**
Nombre de k-mers	2471×10^9	2331×10^9
Nombre de k-mers distincts avant fusion	251×10^9	111×10^9
Nombre de k-mers distincts après fusion	95×10^9	15×10^9
Mémoire (Go)	62	62
Disque (Go)	1661	795
Temps total (min)	1338	862
MKC-Comptage (min)	758	573
MKC-Fusion (min)	148	77
Distances simples (min)	432	212
Distances complexes (min)	8957	4160

Table 3.2. Performances de Simka et statistiques des k-mers sur le projet HMP entier. Simka a été lancé sur une machine équipée d'un processeur Intel E5-2640 de 20 cœurs (2.50 GHz) et 264 Go de mémoire, avec $k = 31$. Le nombre de k-mers distincts a été calculé avant et après l'algorithme MKC-Fusion : le nombre *avant fusion* est obtenu en sommant le nombre de k-mers distincts de chaque jeu de lectures indépendamment, tandis que dans le cas du nombre *après fusion*, les k-mers partagés par plusieurs jeux de lectures sont comptés une seule fois. Ce nombre correspond donc au nombre de vecteurs d'abondances générés. La ligne *Temps total* n'inclue pas le temps de calcul des distances complexes puisqu'il s'agit d'une option de Simka.

métagénomiques, tels que le projet HMP entier.

3.5.2 Performances sur le projet HMP entier

Sur le projet HMP tout entier (690 jeux de lectures), le temps d'exécution total de Simka est de 14 heures avec une empreinte mémoire faible (tableau 3.2). En comparaison, Metafast a dépassé la mémoire de notre machine (Metafast a également manqué de mémoire pour traiter un sous-ensemble du projet HMP composé de 138 jeux de données d'intestin), tandis que Commet a pris plusieurs jours pour calculer les distances d'un jeu de lectures contre tous les autres jeux et aurait donc requis des années de calcul pour obtenir la matrice de distances toute entière. A l'inverse, Mash a fini en moins de 5 heures (255 min) et est plus rapide que Simka. Cela était attendu puisque Mash délivre une approximation de la distance qualitative de Jaccard en se basant sur 10 000 k-mers. Simka, quant à lui, calcule de nombreuses distances, dont celles quantitatives, en se basant sur 15 milliards de k-mers distincts (tableau 3.2).

Ces résultats ont été obtenus avec les paramètres par défaut, c'est à dire en filtrant les k-mers vus une seule fois. Sur ce projet, ce filtre a enlevé seulement 5% des données : les k-mers solides (k-mers vus au moins deux fois) comptent pour 95% des paires de bases du projet entier (tableau 3.2). Mais lorsqu'on regarde en termes de k-mers distincts, les k-mers solides représentent moins de la moitié du nombre de k-mers distincts avant de les fusionner et 15 % des k-mers distincts totaux lorsque ceux-ci sont fusionnés en vecteurs d'abondances. Par conséquent, les performances de Simka, en termes de temps et d'empreinte disque, sont considérablement améliorées lorsque seuls les k-mers solides sont considérés. L'impact de ce filtre sur la qualité des distances est montré dans le chapitre suivant.

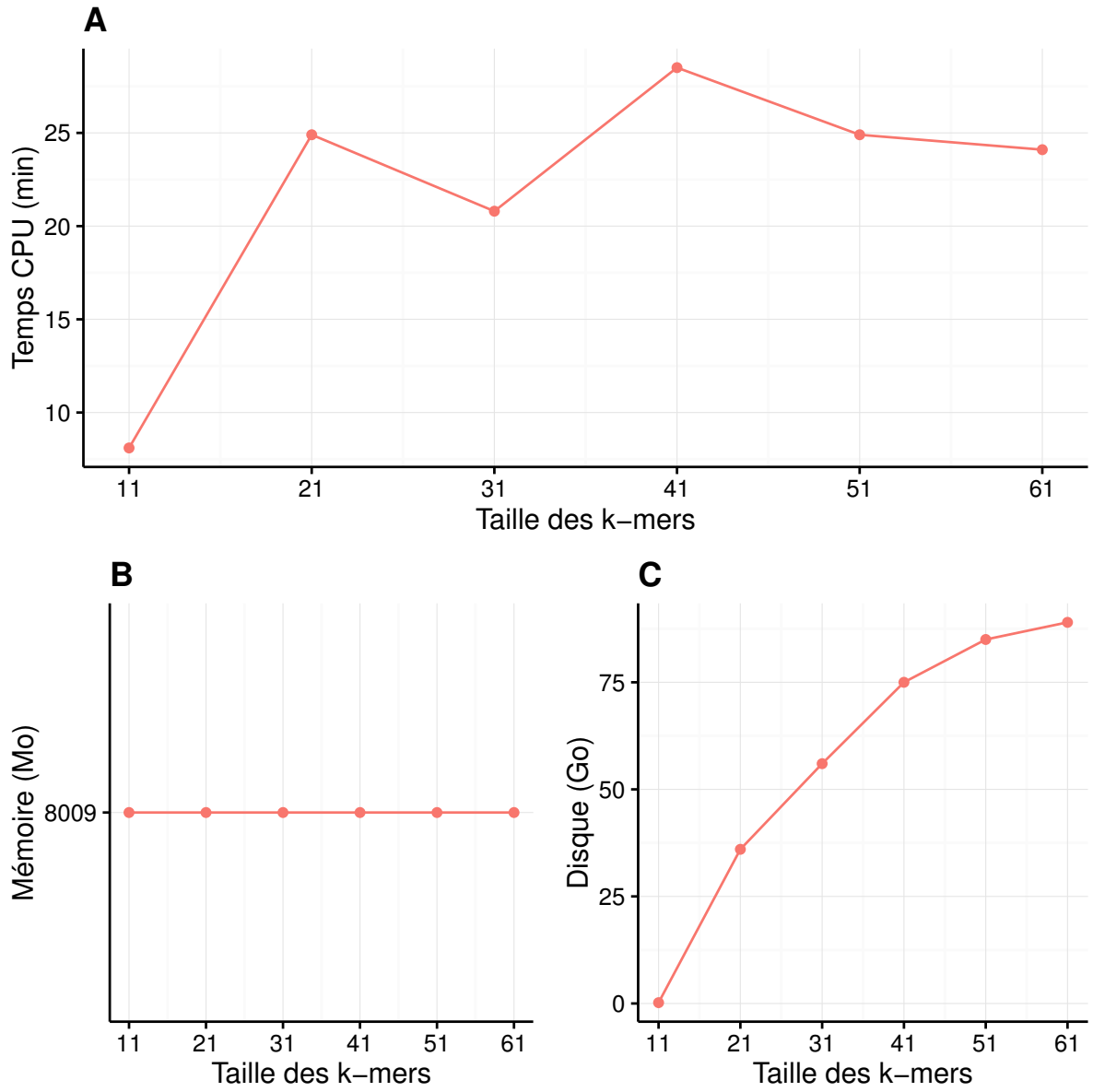

Figure 3.6. Impact de la taille des *k*-mers sur les performances de SIMKA. Ces tests ont été effectués sur les 20 plus gros jeux de lectures du projet HMP. (**A**) Temps CPU par rapport à k. (**B**) Empreinte mémoire par rapport à k. (**C**) Utilisation du disque par rapport à k.

3.5.3 Impact de la taille des *k*-mers

Des tests additionnels ont été effectués pour mesurer l'impact de k sur les performances de Simka sur 20 grands jeux de lectures du projet HMP. La figure 3.6 montre que l'utilisation du disque augmente de manière sous-linéaire avec k. De manière intéressante, le choix de k n'impacte pas l'empreinte mémoire et ne fait que légèrement varier le temps d'exécution total.

3.5.4 Scalabilité de SIMKA

Des tests de parallélisation de Simka ont été effectués sur un cluster composé de 25 nœuds de 8 cœurs chacun. Ces nœuds sont liés à un disque à distance sur lequel les fichiers temporaires

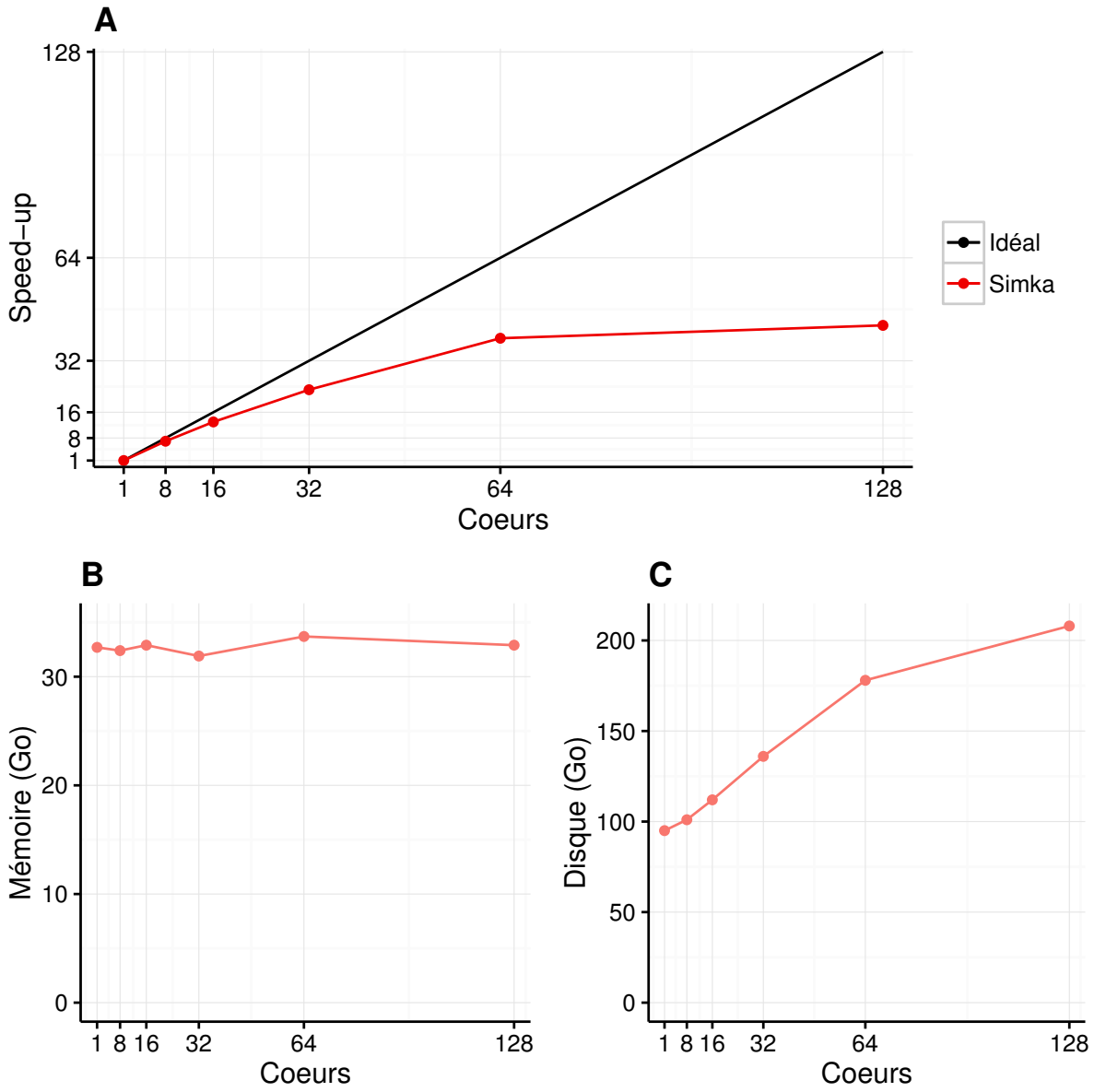

Figure 3.7. Scalabilité de SIMKA. Les performances de Simka sont présentées en fonction du nombre de cœurs. Ces tests ont été effectués sur 100 jeux de données du projet HMP de 5 millions de lectures. Simka a été configuré pour ne pas dépasser 100 Go de mémoire (-max-memory 100000).

de Simka ont été écrits.

La figure 3.7-A indique que Simka bénéficie pleinement de la parallélisation lorsque le nombre de cœurs utilisés est celui d'une machine standard (entre 2 et 16 cœurs). Le gain de temps de calcul est nettement moindre pour 32 et 64 cœurs. Au delà de 64 cœurs, le gain est négligeable. La raison provient de la latence du disque à distance utilisé. Simka est une technique qui repose énormément sur le disque. Il serait intéressant de refaire cette expérimentation sur différentes technologies de stockage (disque local, SSD, etc.). Pendant cette thèse, nous avons notamment eu accès à un super-calculateur sur lequel Simka a été beaucoup plus performant (expérience sur le projet Tara Oceans décrite en section 4.2).

De manière intéressante, la parallélisation n'impacte pas la mémoire requise par Simka

(figure 3.7-B). Pour rappel, la parallélisation de Simka s'obtient en ajustant le nombre de partitions de k-mers, une partition par cœur de calcul. Augmenter le nombre de cœurs augmente donc le nombre de partitions. Cela a pour effet de réduire la taille de chaque partition et donc le pic d'utilisation de la mémoire par la phase de tri du MKC.

L'utilisation du disque augmente avec le nombre de cœurs (figure 3.7-C) car Simka a été paramétré pour utiliser un processus de comptage par coeur, chacun écrivant des fichiers temporaires pour compter les k-mers.

3.6 Conclusion

Ce chapitre a présenté Simka, une nouvelle méthode *de novo* pour calculer une collection de distances entre de nombreux jeux de données métagénomiques en se basant sur leur composition en k-mers. Cela a été possible grâce au développement du compteur de k-mers multi-jeux (MKC), une nouvelle stratégie qui compte très efficacement les k-mers en termes de temps, mémoire et disque. La nouveauté de cette stratégie est qu'elle compte simultanément les k-mers de n'importe quel nombre de jeux de lectures et représente les résultats comme un flux de données, fournissant les comptages dans tous les jeux de données, k-mer par k-mer.

À ce jour, seul l'outil Mash a de meilleurs performances que Simka. Cependant, celui-ci est limité au calcul d'une estimation de l'index de Jaccard. Dans le chapitre suivant, qui évalue la qualité des distances basées sur les k-mers, nous constatons que son usage est restreint à un nombre limité de situations.

Chapitre 4

Évaluation de la qualité des distances calculées par Simka

Dans ce chapitre, nous évaluons tout d'abord la qualité des distances calculées par Simka. Dans un second temps, nous détaillons une application réelle dans le cadre du projet Tara Oceans [3].

Dans la section 4.1, d'une part la qualité des distances est évaluée en les comparant à des distances *de novo*, puis à des distances traditionnelles basées sur la composition taxonomique des jeux de lectures et enfin à des résultats biologiques connus. L'impact des paramètres importants de Simka (taille des k-mers, filtre d'abondance) sur les distances est également mesuré. La section 4.2 présente les résultats d'une étude du consortium Tara Oceans dans laquelle nous avons été impliqués. Il s'agit d'une analyse comparative à l'échelle de la planète basée sur les résultats de Simka.

4.1 Évaluation des distances

La plupart des expérimentations a été conduite sur les données du Human Microbiome Project (HMP) [1]. C'est actuellement un des plus gros projets métagénomiques en terme de jeux de lectures plein-génome : 690 échantillons de différents tissus (`http://www.hmpdacc.org/HMASM/`). Le projet entier a été séquencé via la technologie Illumina et contient en tout 2×16 milliards de lectures appairées d'une centaine de nucléotides réparties non uniformément à travers les 690 jeux de données. Un des avantages de ce projet est que ses données ont été largement étudiées (voir `http://hmpdacc.org/pubs/publications.php` pour une liste complète). En particulier, les communautés microbiennes sont relativement bien représentées dans les bases de données de références [1, 46].

Nous avons évalué la qualité des distances calculées par Simka en répondant à deux questions. Premièrement, sont-elles similaires à des distances entre jeux de lectures calculées avec d'autres approches ? Deuxièmement, est-ce qu'elles retrouvent des structures connues des jeux du projet HMP ? Pour la première évaluation, les résultats de Simka ont été comparés à deux types de distance : des distances *de novo* basées sur des comparaisons de lectures et des distances taxonomiques, c'est-à-dire basées sur la composition taxonomique des jeux.

4.1.1 Corrélation avec des approches *de novo* basées sur des comparaisons de lectures.

Simka a été comparé à deux approches *de novo* basées sur des comparaisons de lectures : Commet [107] et une méthode d'alignement en utilisant l'outil Blat [103]. Ces deux approches

définissent et utilisent une notion de similarité entre les lectures. Elles dérivent le pourcentage de lectures de S_i similaires à au moins une lecture de S_j, noté $\left|S_i \overrightarrow{\cap} S_j\right|$. COMMET considère que deux lectures sont similaires si elles partagent au moins t k-mers non chevauchants (ici $t = 2$, $k = 33$). Pour les alignements de BLAT, nous considérons que deux lectures sont similaires si leur alignement est au moins d'une taille de 70 nucléotides avec un pourcentage d'identité supérieur à un seuil. Nous avons utilisé 3 seuils d'identité : 92%, 95% et 98%. La similarité globale entre deux jeux de lectures S_i et S_j fournie par ces deux approches est donc donnée par :

$$PourcentageLecturesCommunes(S_i, S_j) = 100 \times \frac{\left|S_i \overrightarrow{\cap} S_j\right| + \left|S_j \overrightarrow{\cap} S_i\right|}{|S_i| + |S_j|}. \tag{4.1}$$

Pour comparer SIMKA à ces approches basées sur les lectures, nous utilisons une mesure de similarité définie par le pourcentage de k-mers en commun entre deux jeux de lectures S_i et S_j :

$$PourcentageKmersCommuns(S_i, S_j) = 100 \times \frac{\sum_{w \in S_i \cap S_j} N_{S_i}(w) + N_{S_j}(w)}{\sum_{w \in S_i} N_{S_i}(w) + \sum_{w \in S_j} N_{S_j}(w)} \tag{4.2}$$

Cela est l'équivalent de l'équation 4.1 mais basé sur les k-mers. Les deux équations ci-dessus sont une estimation du contenu génomique partagé par deux jeux de lectures au niveau des lectures et au niveau des k-mers respectivement.

Ces mesures ont été calculées par SIMKA et COMMET sur les 50 plus petits jeux de lectures du projet HMP. Pour des raisons de temps de traitement, BLAT et SIMKA ont été comparés sur un plus petit sous-ensemble : les 15 plus petits jeux de lectures.

Corrélation avec les résultats de COMMET. Regarder la similarité avec COMMET est intéressant car cet outil utilise une heuristique basée sur les k-mers partagés mais sa distance finale est exprimée en termes de nombre de lectures. Comme montré par la figure 4.1, les mesures de similarité de SIMKA et COMMET sont extrêmement corrélées (coefficient de corrélation de Spearman $r = 0.989$).

Corrélation avec les résultats de BLAT. De manière similaire, une nette corrélation ($r > 0.89$) est également observée entre le pourcentage de k-mers partagés et le pourcentage de lectures similaires détectées par BLAT (figure 4.2). De façon intéressante, la corrélation dépend de la taille des k-mers et du seuil d'identité utilisé par BLAT : les plus grandes tailles de k-mers corrèlent mieux avec les plus grands seuils d'identité et inversement. La plus grande valeur de corrélation est de 0.987, obtenue par SIMKA avec $k = 21$ comparé aux résultats de BLAT avec 95% d'identité.

Ces résultats démontrent que les métriques basées sur des comparaisons de lectures peuvent être remplacées sans danger par celles basées sur des k-mers et cela permet un gain de temps énorme sur de grands projets métagénomiques. De plus, la taille des k-mers joue un rôle similaire au seuil d'identité des méthodes basées sur de l'alignement et permet d'ajuster le niveau de précision avec lequel les communautés sont comparées.

4.1.2 Corrélation avec des distances taxonomiques sur les données d'intestin.

Une manière traditionnelle pour comparer les échantillons métagénomiques est de calculer des distances taxonomiques qui sont basées sur l'assignation des lectures à des taxons en les alignant sur des références. Pour comparer SIMKA à une telle approche, nous avons utilisé les échantillons de flore intestinale du projet HMP car c'est un milieu très étudié contenant 138 jeux de données. Le consortium HMP fournit également le profil taxonomique quantitatif de

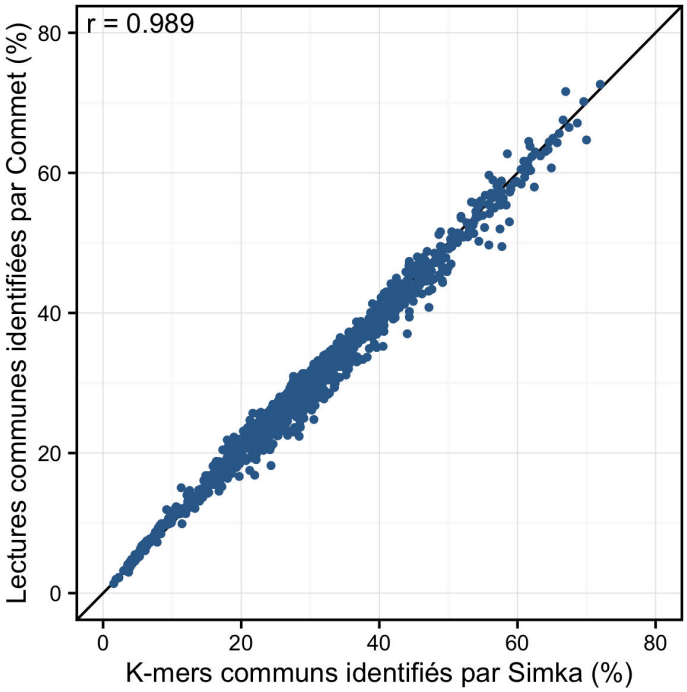

Figure 4.1. Comparaison des mesures de similarité de Simka et Commet. Commet et Simka ont été lancés avec la valeur de k par défaut de Commet ($k = 33$). Sur ce nuage de points, chaque point représente une paire de jeux de lectures, dont la coordonnée X est le % de k-mers partagés calculé par Simka, et la coordonnée Y est le pourcentage de lectures similaires calculé par Commet.

chaque jeu sur son site web (`http://www.hmpdacc.org/HMSCP/`). Ces profils ont été obtenus en alignant les lectures sur un catalogue de génomes de référence à 80% d'identité. Le package R *vegan* a ensuite été utilisé pour dériver des distances écologiques de ces profils. Seuls Mash et Simka ont été considérés pour cette expérimentation. Comme mentionné précédemment, Commet et Metafast ne passent pas à l'échelle sur ce jeu de données. Nous n'espérons pas une corrélation parfaite entre les distances *de novo* et taxonomiques car elles ne considèrent pas la même quantité de données. La figure 4.3 montre qu'une forte fraction de lectures ne s'aligne pas sur les références quel que soit le tissu considéré. À l'inverse, les méthodes *de novo* utilisent toutes les lectures.

Les résultats de Simka et Mash ont été comparés à des distances taxonomiques qualitatives et quantitatives. Pour la comparaison qualitative, nous avons utilisé l'index de Jaccard car il est fournit par Simka et Mash. Pour la comparaison quantitative, nous avons comparé la distance de Bray-Curtis de Simka à une distance de Bray-Curtis taxonomique. En revanche, Mash ne fournit pas d'indices quantitatifs. Pour comparer de manière juste son index de Jaccard, nous avons utilisé une distance de Jaccard taxonomique prenant en compte l'abondance des espèces (appelée *Weighted Jaccard* ou *Jaccard bag similarity*). Mash a été lancé avec les mêmes paramètres que dans sa publication lors de son propre traitement du projet HMP, c'est-à-dire en utilisant 10 000 k-mers par jeu de lectures pour estimer ses index de Jaccard. Simka et Mash utilisent une taille de k-mers de 31.

Les distances quantitatives de Simka apparaissent vraiment bien corrélées aux distances taxonomiques traditionnelles (corrélation de Spearman $r = 0.88$, figure 4.4-A). Sur cette figure, on peut aussi remarquer que les mesures de Simka sont robustes sur toute la gamme de distances représentées. A l'inverse, les distances de Mash ne sont pas bien corrélées avec les distances

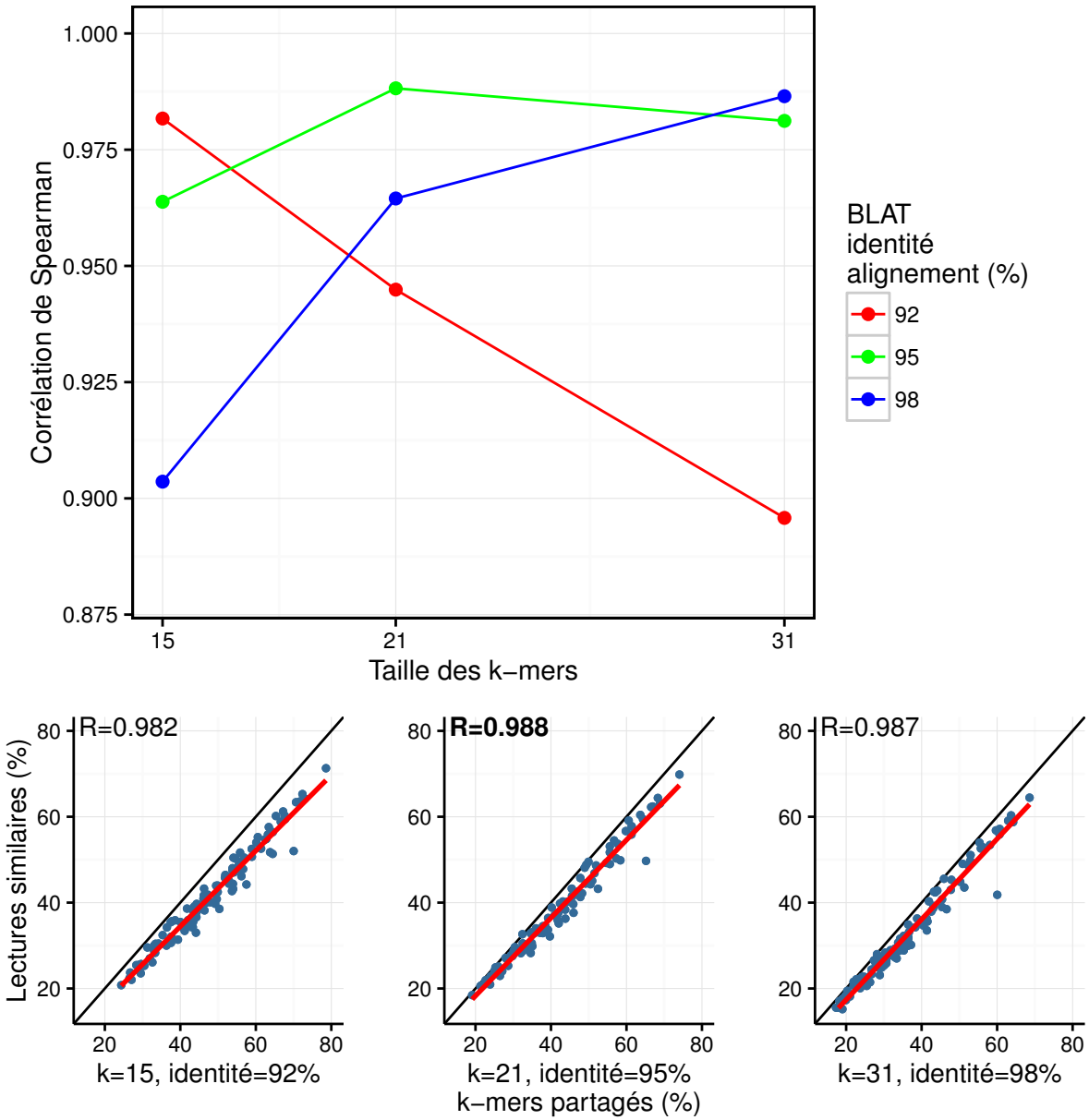

FIGURE 4.2. Comparaison des résultats de Simka et de BLAT pour différentes valeurs de k et plusieurs seuils d'identité de BLAT. Le coefficient de corrélation de Spearman entre les résultats de SIMKA et de BLAT est représentée par rapport à k. Les trois nuages de points montrent précisément la corrélation entre les résultats de SIMKA et BLAT pour les trois valeurs de k considérées ayant la plus haute corrélation.

taxonomiques quantitatives ($r = 0.546$, figure 4.4-B). Cela est probablement dû au fait que les échantillons d'intestin diffèrent plus en termes d'abondances relatives des microbes qu'en terme de présence-absence. Puisque MASH ne peut que fournir des distances qualitatives, il n'est pas équipé pour gérer ce cas de figure.

La figure 4.4-C montre que cette conclusion vaut pour d'autres jeux de données correspondant à d'autres tissus. Dans cette expérimentation, seuls les tissus générant des distances taxonomiques de qualité ont été conservés. Nos exigences de qualité sont les suivantes : (1) le jeu de données doit contenir plus de 10 échantillons pour produire une corrélation de Spearman

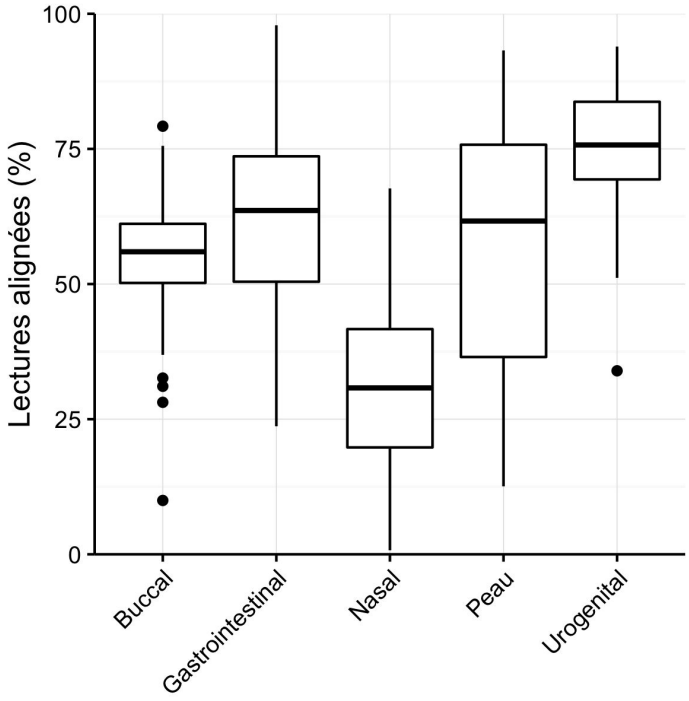

Figure 4.3. Pourcentage de lectures alignées sur des références par tissu. Les lectures de 690 jeux ont été alignées sur un catalogue de génomes de référence à 80% d'identité par le consortium HMP (`https://www.hmpdacc.org/hmp/doc/ReadMapping_SOP.pdf`).

fiable ; (2) le jeu de données doit avoir plus de 40% de lectures alignées pour que les distances taxonomiques soient jugées fiables et (3) les distances taxonomiques doivent être bien réparties dans la gamme [0, 1]. En particulier, le jeu de données nasal a été enlevé car il n'avait que 33% de lectures assignées à des références. Le jeu de données urogénital a été écarté car il avait une surreprésentation de très grandes distances taxonomiques dans la gamme [0.99-1] (médiane=0.97), signifiant que la corrélation n'aurait été mesurée que sur une petite gamme de distances.

Les distances qualitatives de Simka et Mash sont toutes les deux très mal corrélées aux distances taxonomiques qualitatives ($r < 0.51$, figure 4.5-A-B)). Une hypothèse peut être formulée ici. La qualité des distances taxonomiques qualitatives doit être largement moins bonne que celle des distances quantitatives. En effet, une distance quantitative donne plus de poids aux espèces abondantes. Ces espèces abondantes doivent être bien représentées dans les banques de références puisque leur génome est bien couvert. À l'inverse, une distance qualitative est impactée équitablement par les espèces abondantes et rares. Or les espèces rares sont probablement mal représentées dans les banques car il est plus complexe de les assembler. Une large partie des lectures non alignées doit provenir de ces espèces rares. Or ces lectures sont prises en compte par Mash et Simka mais pas par une analyse taxonomique.

4.1.3 Impact des paramètres de Simka

L'impact des deux paramètres importants de Simka sur ses distances a été mesuré sur les 138 jeux de lectures d'intestin du projet HMP : la taille des k-mers k et le filtre d'abondance. Pour rappel, pour un jeu donné, le filtre d'abondance supprime tous les k-mers vus une seule fois. Les k-mers restants sont dit "solides". Pour cela, la corrélation de Spearman entre les résultats de Simka et les distances taxonomiques entre les 138 jeux ont été mesurées pour différentes valeurs de k et en activant ou non le filtre d'abondance. Cette expérimentation a été effectuée sur une

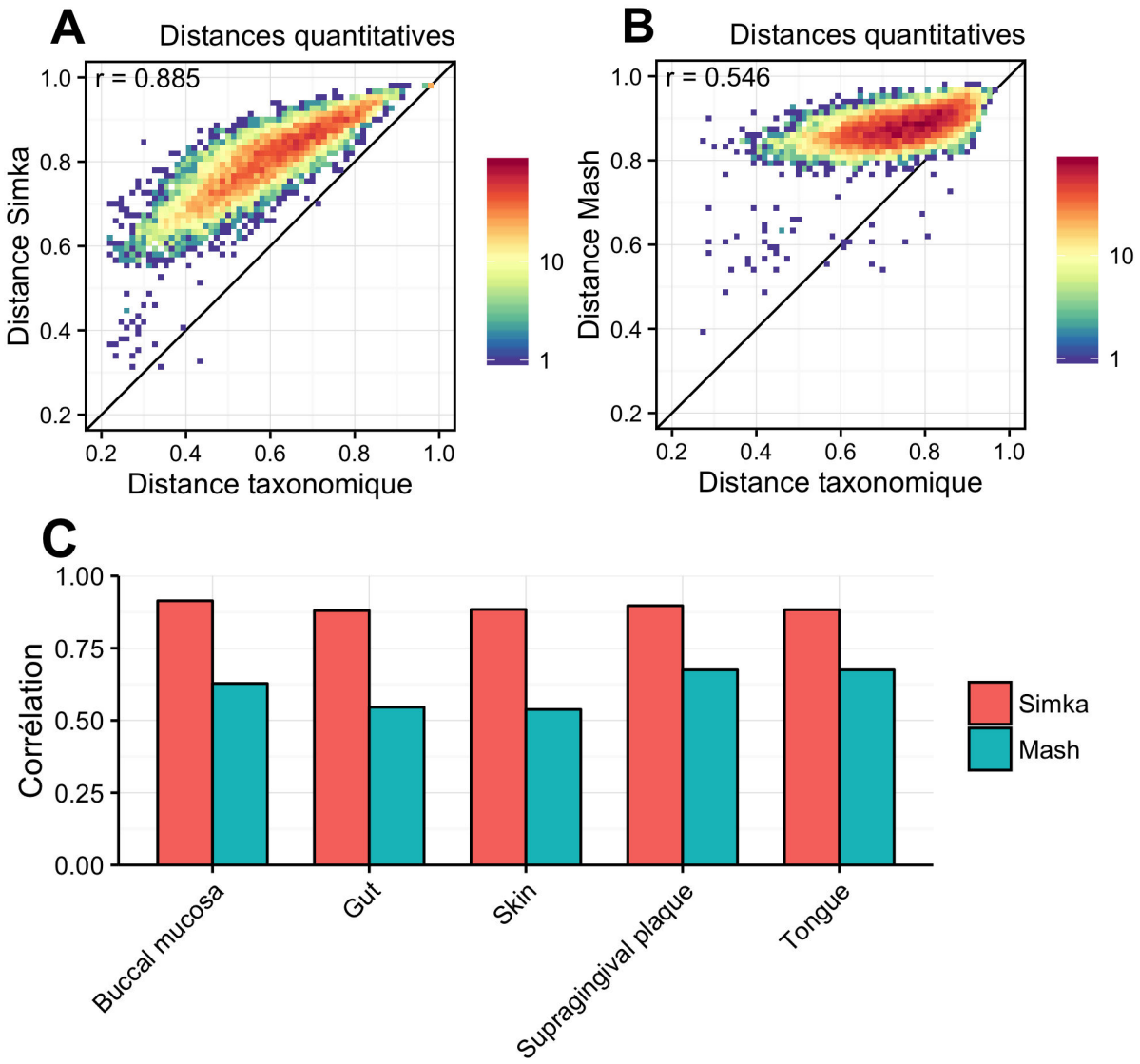

Figure 4.4. Corrélation entre des distances quantitatives taxonomiques et *de novo*. Sur les figures de densité, chaque point représente une ou plusieurs paires d'échantillons d'intestin du projet HMP. La coordonnée X indique la distance de taxonomique, et la coordonnée Y la distance *de novo*. La couleur d'un point est fonction de la quantité de paires d'échantillons ayant la paire de distances données (échelle logarithmique). (**A**) Corrélation avec la distance de Bray-Curtis de Simka. (**B**) Corrélation avec l'index de Jaccard de Mash. (**C**) Résultats de corrélations de Spearman obtenues par Mash et Simka sur d'autres tissus.

distance quantitative et qualitative de Bray-Curtis.

Comme on peut le voir sur la figure 4.6-A, lorsque l'on considère une distance quantitative, la corrélation est très bonne et stable selon le choix de k dès que k est plus grand que 15, avec un optimum à 21. Notamment, pour de très faibles valeurs de k ($k < 15$), la corrélation chute ($r = 0.5$ pour $k = 11$). Cela complète une étude précédente qui suggérait que plus grande est la taille des k-mers, meilleure est la corrélation avec des distances taxonomiques [130]. Cette étude était cependant limitée à une taille de k-mer inférieure à 13. Le filtre d'abondance, quant à lui, n'a quasiment aucun impact. Comme montré dans le tableau 3.2, même si le filtre d'abondance jette 85% des k-mers distincts ; les k-mers solides représentent encore 95% des paires de bases

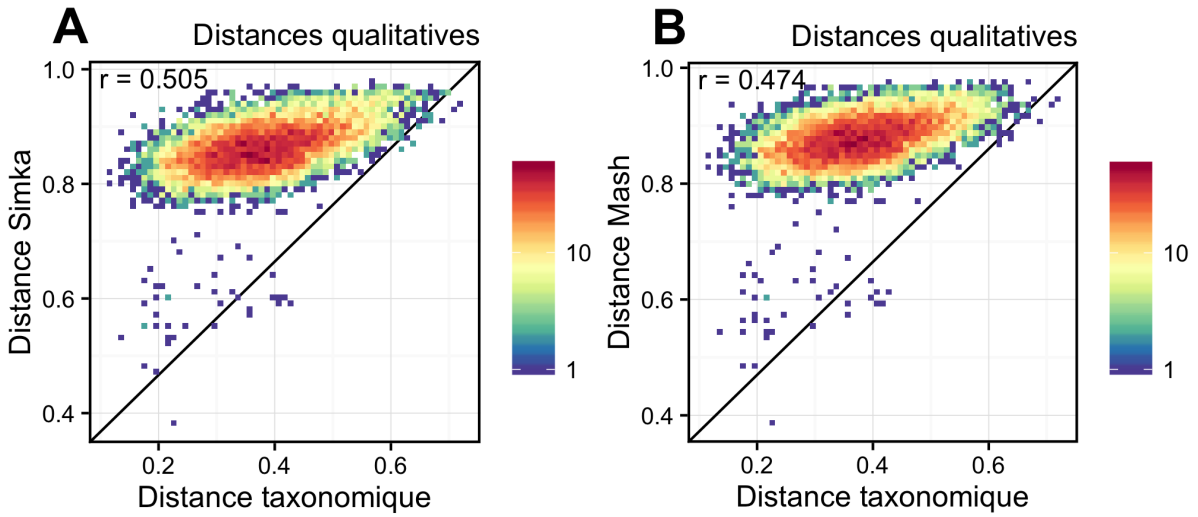

Figure 4.5. Corrélation entre des distances qualitatives taxonomiques et *de novo*. Toutes les distances utilisées sont des distances qualitatives de Jaccard.

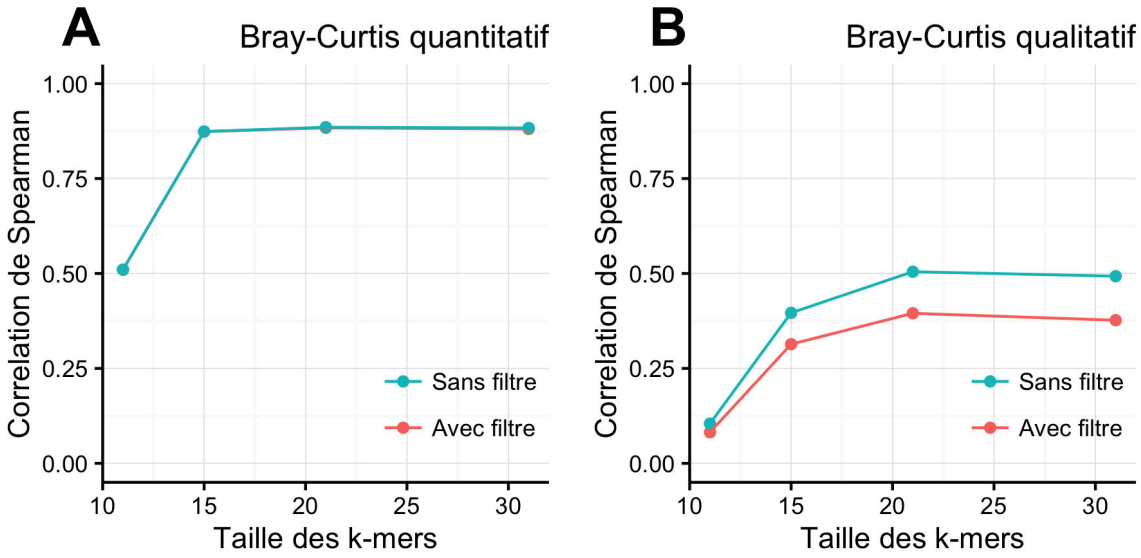

Figure 4.6. Impact de la taille des *k*-mers et du filtre d'abondance sur la corrélation avec des distances taxonomiques. Sur les 138 jeux de lectures d'intestin du projet HMP, la corrélation de Spearman entre les résultats de Simka et leur distance taxonomique a été mesurée pour différentes valeurs de k et en activant ou non le filtre d'abondance. Les distances de Simka et taxonomiques utilisées sont (**A**) la distance quantitative de Bray-Curtis et (**B**) la distance qualitative de Bray-Curtis. Les courbes sont superposées sur la figure de gauche.

du projet HMP, ce qui explique la faible différence.

Comme nous l'avons vu précédemment, la corrélation est beaucoup plus faible pour les distances qualitatives (figure 4.6-B). L'optimum reste cependant à $k = 21$ comme pour les distances quantitatives. Les distances qualitatives n'étant impactées que par la présence-absence des k-mers distincts, elles sont plus sensibles au filtre d'abondance.

4.1.4 Visualisation de la structure des jeux de lectures du projet HMP

Nous proposons de visualiser la structure des jeux de données du projet HMP afin de voir si Simka est capable de reproduire des résultats biologiques connus. Pour visualiser facilement ces structures, nous avons utilisé l'analyse en coordonnées principales (PCoA) [131], aussi couramment appelé *Multidimensional scaling* (MDS), afin d'obtenir une représentation des jeux de lectures et leurs distances sur un plan 2D. Cette représentation 2D préserve les valeurs originales de la matrice de distances de manière optimale.

La figure 4.7 montre plusieurs PCoA de différentes distances calculées par Simka sur tous les jeux de lectures du projet HMP. Pour rappel, les définitions des distances sont données dans le tableau 3.1. La figure 4.7-A (distance quantitative de Ochiai) indique que les jeux de lectures se séparent clairement par tissu. Ce résultat est en adéquation avec les résultats d'autres études du consortium HMP [1, 132, 127]. De plus, on peut remarquer que différentes distances peuvent mener à différentes distributions des échantillons (figure 4.7-B-C-D), où l'on peut voir que certains groupes sont plus ou moins discriminés. Cela confirme qu'il est important de conduire les analyses en utilisant plusieurs distances comme suggéré dans [127, 86], puisque différentes distances peuvent capturer des caractéristiques différentes des jeux de données.

Nous avons effectué la même expérience sur les 138 échantillons d'intestin afin de rechercher un résultat biologique plus fin. Arumugam *et al.* [89] ont montré que les échantillons d'intestin humain sont organisés en trois groupes, nommés entérotypes, caractérisés par l'abondance de quelques genres : *Bacteroides*, *Prevotella* et les genres de la famille *Ruminococcaceae*.

Les entérotypes originaux ont été construits à partir de la distance de Jensen-Shannon sur des profils taxonomiques. La figure 4.8 montre la PCoA de la distance de Jensen-Shannon calculée par Simka. Nous avons ajouté l'information de l'abondance relative des 3 entérotypes dans chaque jeu par un gradient de couleurs. Cela révèle que les échantillons sont clairement distribués en fonction de l'abondance de ces trois genres. Les distances de Simka retrouvent donc une structure biologique dont on n'avait aucune connaissance *a priori* : ici, le fait que les échantillons d'intestin sont structurés le long de gradients d'abondances de *Bacteroides*, *Prevotella* et *Ruminococcaceae*.

4.1.5 Résultats sur un environnement complexe

Le microbiome humain n'est pas représentatif de toutes les communautés d'organismes existantes. Les environnements aquatiques et terrestres sont connus pour être plus complexes [133]. Malheureusement, ils sont moins étudiés que l'humain. Il n'est pas possible actuellement d'effectuer une analyse avec références fiable. En revanche, des analyses comparatives *de novo* d'échantillons complexes ont déjà été effectuées en utilisant des outils conventionnels, tels que Blast. C'est le cas du projet océanique Global Ocean Sampling (GOS).

Le projet GOS met à disposition 44 jeux de lectures océaniques, contenant chacun en moyenne 174 759 lectures (1249 nucléotides par lecture en moyenne séquencés avec la technologie Sanger), que nous avons comparés avec Simka. Les comparaisons ont été effectuées avec et sans filtre d'abondance afin de voir l'effet de ce filtre sur des données peu couvertes. Les deux matrices de distances résultantes ont été représentées par des heatmaps dont les colonnes et lignes ont été arrangées grâce à un clustering hiérarchique (figure 4.9), de la même manière que dans la publication de GOS. Cette figure montre deux points importants. Premièrement, Simka parvient à retrouver la structure des échantillons par océan. Il arrive donc aux mêmes conclusions que le consortium GOS qui est que les échantillons plus proches géographiquement ont une biodiversité plus proche et vice versa. Deuxièmement, le filtre d'abondance n'a quasiment aucun impact sur le clustering des jeux de données. Cela a été confirmé par une valeur de corrélation de Spearman de 0.981 entre la matrice de distance filtrée et celle non filtrée. Nous n'avons pas

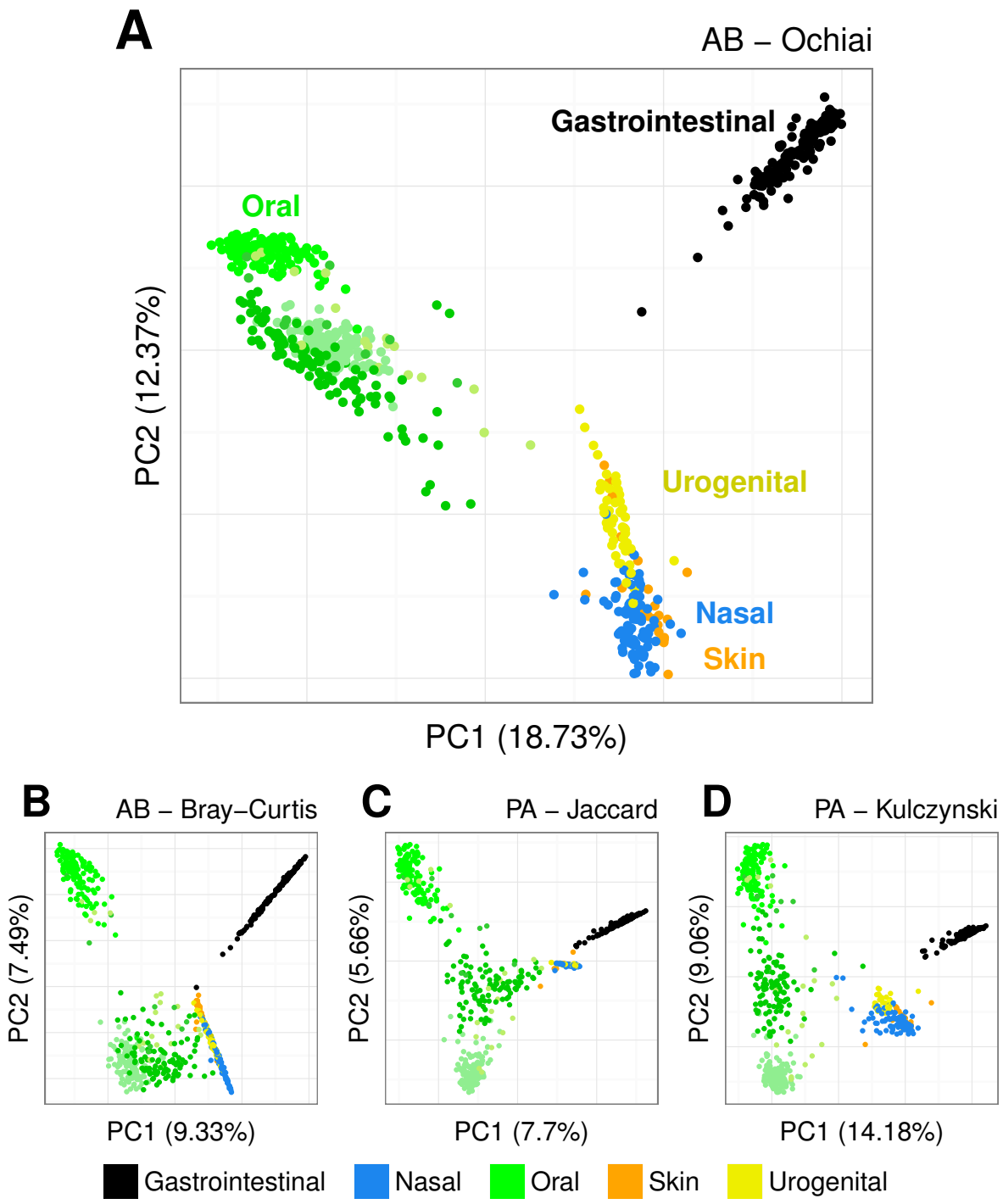

Figure 4.7. Distribution de la diversité des jeux de lectures du projet HMP par tissus. Chaque figure correspond à une PCoA des jeux de lectures obtenue par une distance donnée calculée par Simka avec $k = 21$. AB et PA indique respectivement que la distance est quantitative ou qualitative. Chaque point correspond à un échantillon et est coloré par le tissu humain d'où il a été extrait. Les nuances de vert correspondent à 3 différents sous-tissus des échantillons oraux : langue (dorsum), planque dentaire (supragingival), muqueuse buccale (Buccal mucosa).

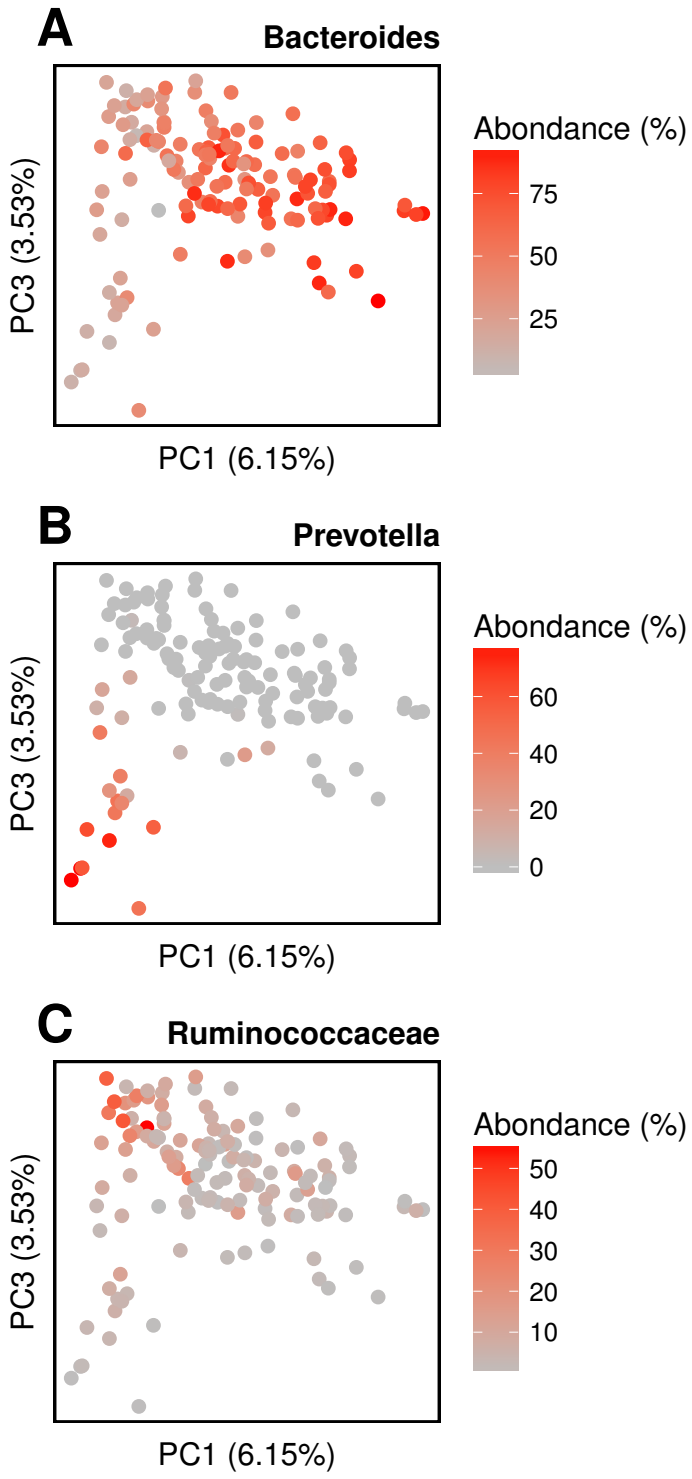

FIGURE 4.8. Abondances relatives des genres principaux des échantillons d'intestin du projet HMP. La distribution des échantillons d'intestin du projet HMP est montrée via une PCoA de la distance de Jensen-Shannon. La matrice de distance a été calculée par SIMKA avec $k = 21$. Les abondances relatives (0-100%) de (**A**) *Bacteroides*, (**B**) *Prevotella* et (**C**) *Ruminococcaceae*, calculées avec le logiciel Metaphlan [43], ont été représentées comme des nuances de couleurs sur les points.

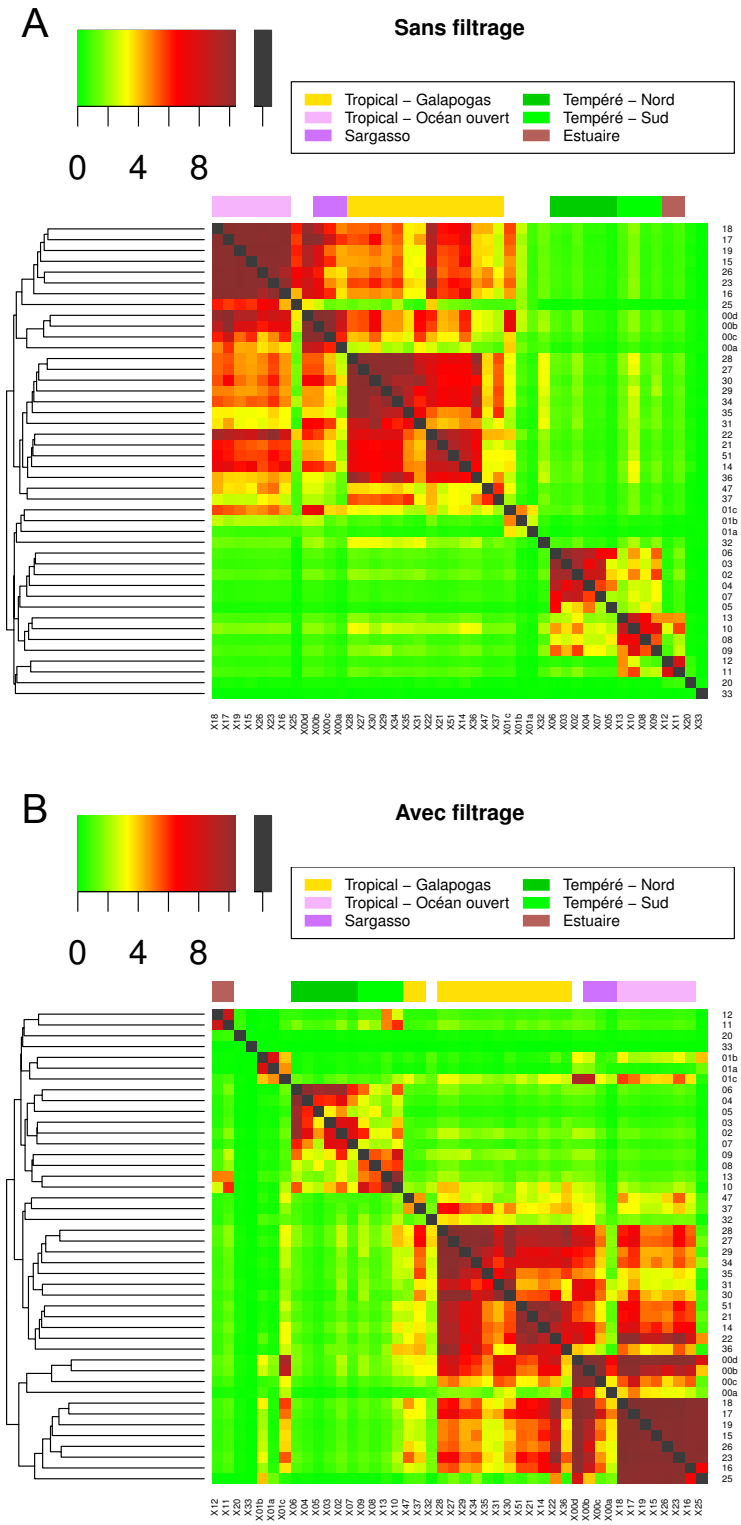

Figure 4.9. Résultats de Simka sur des jeux de données peu couverts du projet Global Ocean Sampling (GOS) Comparaison du clutering obtenu par Simka en utilisant la distance quantitative de Bray-Curtis (**A**) sans filtre d'abondance et (**B**) en filtrant les k-mers vus une seule fois.

noté de différences majeures entre les distances qualitatives et quantitatives sur ces données.

4.2 Application aux données de Tara Oceans

SIMKA a été élaboré pour passer à l'échelle sur de très grands projets métagénomiques provenant possiblement d'environnements complexes. Pendant cette thèse, nous avons collaboré avec le consortium Tara Oceans pour effectuer une analyse comparative à l'échelle de la planète en utilisant l'ensemble de leurs données métagénomiques. Notre travail a consisté à fournir les données d'entrée de cette étude, à savoir les matrices de distances de SIMKA. L'ensemble des analyses que nous présentons ont été effectuées par différents laboratoires du consortium Tara Oceans.

L'objectif est d'établir la première biogéographie à partir de données métagénomiques des communautés virales, bactériennes et eucaryotes dans les eaux océaniques de surface et d'enquêter sur les différents facteurs qui peuvent impacter l'organisation globale des communautés planctoniques.

Données. Dans cette étude, 644 échantillons métagénomiques plein-génome ont été considérés, chacun contenant des centaines de millions de lectures. Le nombre total de lectures s'élève à 240 milliards (∼15 To de données compressées). Les échantillons ont été prélevés à une centaine de positions géographiques différentes, appelées stations. Deux profondeurs de prélèvement ont été considérées : à la surface de l'eau (SUR) et à la profondeur où il y a un maximum de concentration de chlorophylle (DCM). Les échantillons sont répartis de manière uniforme en 6 fractions de taille d'organismes : 0 à 0.2 µm (virus), 0.22 à 3 µm (bactéries), 0.8 à 5 µm (protistes), 5 à 20 µm (métazoaires), 20 à 180 µm (métazoaires) et 180 à 2000 µm (métazoaires).

Des jeux d'amplicons ont également été séquencés à partir des mêmes échantillons. Il a été nécessaire de faire des analyses conventionnelles basées sur des OTUs pour s'assurer d'une cohérence minimale des résultats fournis par SIMKA. En effet, SIMKA (et plus généralement la métagénomique comparative basée sur des k-mers) est un nouvel outil et n'est pas encore adopté par toute la communauté scientifique. Cette étude a permis de mesurer le lien entre les distances de SIMKA (calculées à partir de données plein-génome) et les distance basées sur des OTUs (calculées à partir de jeux de données ciblées).

Infrastructure de calcul. Le supercalculateur Airain du Très Grand Centre de Calcul du CEA (TGCC) a été utilisé pour effectuer les calculs de distances. Airain dispose d'un espace de stockage de 2.3 Po sur lequel sont hébergées les données de Tara Oceans. Cet espace de stockage est intéressant car il dispose d'un système de gestion de fichier distribué, nommé *Lustre*, capable de fonctionner sur plusieurs centaines de nœuds sans altérer sa vitesse. Nous avons donc pu déployer SIMKA sur environ 500 cœurs de calcul sans impact notable concernant les écritures disques.

Exécutions de SIMKA. Une exécution de SIMKA a été effectuée par fraction de taille d'organismes, en considérant donc une centaine de jeux de données à la fois. Il n'y a pas eu besoin de lancer SIMKA sur l'ensemble des jeux, toutes fractions confondues, car elles ne partagent pas les mêmes organismes en théorie. Nous avons choisi une taille de k-mer de 31 pour qu'ils soient bien spécifiques des espèces dans cet environnement complexe. Le filtre d'abondance n'a pas été activé. En effet, nous avons détecté qu'environ 80% des k-mers distincts ne sont vus qu'une seule fois. De plus, ces k-mers comptent pour 50% des paires de bases totales. En comparaison, sur le projet HMP, les k-mers vus une seule fois ne comptaient que pour 5% des données. Jusqu'à 500

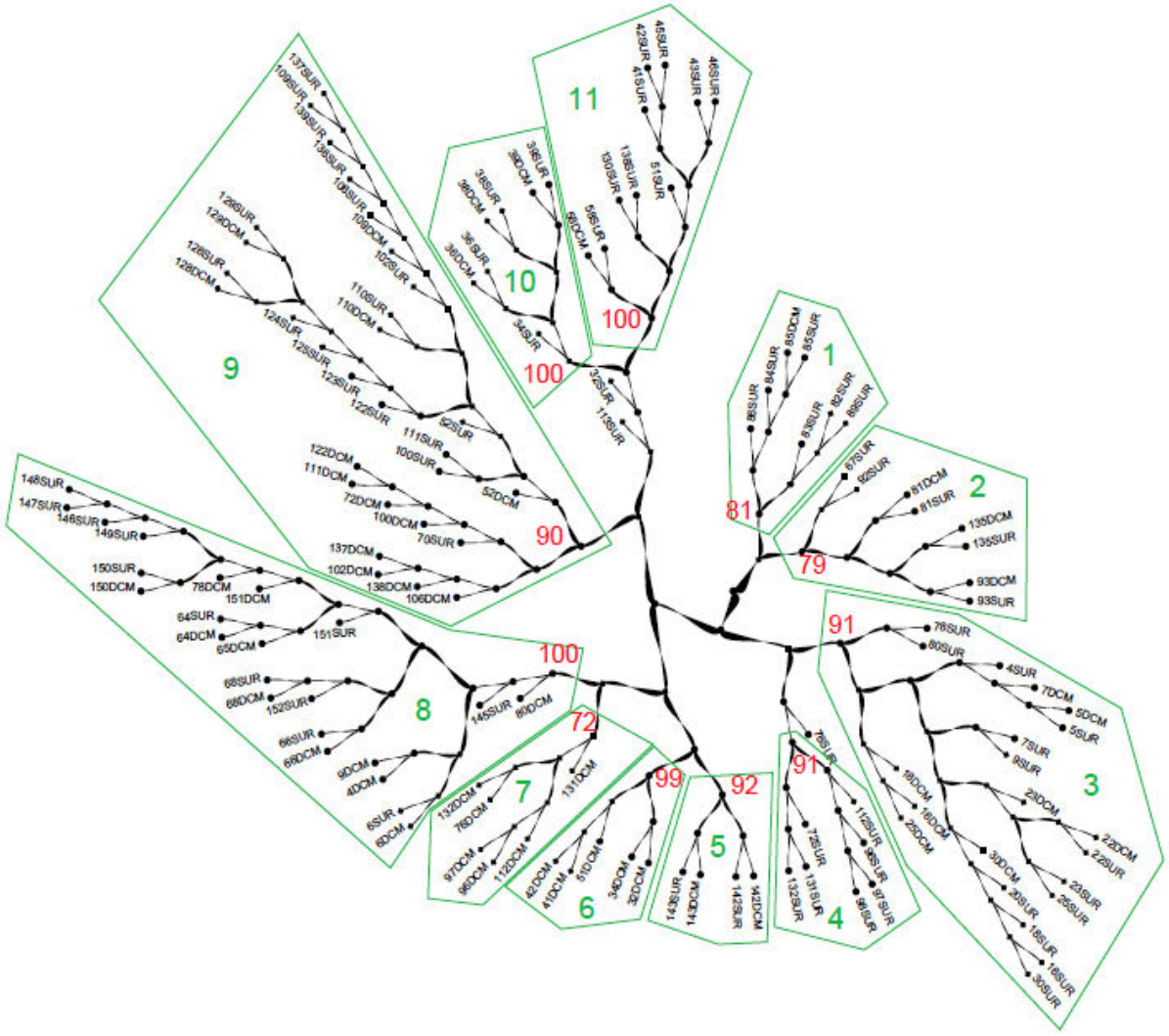

Figure 4.10. Partitionnement des échantillons de Tara Oceans en génocénoses. Dendrogramme radial résultant du clutering UPGMA pour la fraction d'organismes de 0.8 à 5 µm (protistes). Chaque feuille correspond à un échantillon métagénomique. Les valeurs en rouge indique le support des nœuds qui séparent chaque génocénose des autres.

milliards de k-mers distincts ont été considérés par fraction pour un total d'environ 1 trillion de k-mers, soit une abondance moyenne des k-mers très faible. L'étape de comptage des k-mers a requis environ 3h par fraction. Les matrices de distances ont été calculées en seulement 1h. L'ensemble des résultats a été obtenu aussi rapidement que lors de notre test sur le projet HMP entier, alors que le projet Tara Oceans est environ 7 fois plus volumineux en termes de paires de bases. Les efforts que nous avons mis en œuvre pour paralléliser Simka ont pu être exploités pleinement sur cette infrastructure haute performance.

Analyses des résultats de SIMKA. Parmi les distances calculées par Simka, la distance qualitative de Bray-Curtis a été utilisée pour les analyses. Il a été important de travailler en présence-absence car les données de Tara Oceans ont été récoltées sur une longue période. Certains paramètres extérieurs peuvent faire varier l'abondance des organismes, tels que les changements saisonniers.

Sur la base de cette matrice de distances, les échantillons métagénomiques ont été clusteri-

sés hiérarchiquement par la méthode UPGMA (unweighted pair group method with arithmetic mean [134]). La figure 4.10 montre l'arbre résultant de cette opération sur la fraction d'organismes de 0.8 à 5 µm (protistes). En plus de la topologie de l'arbre, des valeurs de support ont été calculées pour chaque nœud. Elles indiquent à quel point les descendants d'un nœud sont stables. Pour les calculer, de nombreuses passes de clustering ont été effectuées en retirant à chaque itération des échantillons de la matrice de distances. Ces valeurs de support ont été prises en compte pour sectionner l'arbre et partitionner les échantillons. Les clusters résultants sont appelés génocénoses comme référence aux biocénoses qui correspondent à l'ensemble des êtres vivants coexistants dans un espace écologique donné.

Lorsque l'on regarde la position géographique de chaque échantillon et leur classification sur une carte du monde, on voit que les génocénoses forment des provinces parfaitement cohérentes d'un point de vue géographique (figure 4.11) et cohérentes avec d'autres études documentant des patterns en biogéographie marine [135, 136, 137]. Sur cette carte, la couleur des pastilles indique la proximité génomique des échantillons. Pour obtenir ces couleurs, la matrice de distances de Simka a tout d'abord été résumée en 3 dimensions grâce à une PCoA. Un échantillon a donc 3 coordonnées. Ces coordonnées ont été transformées linéairement dans l'intervalle [0, 255]. Les trois coordonnées d'un échantillon sont alors utilisées pour encoder les canaux d'une couleur RGB (rouge, vert, bleu), une coordonnée par canal. Cette indication continue de la similarité entre les échantillons a été ajoutée afin de confirmer la robustesse du clustering discret des génocénoses. De plus, elle permet de voir des variations génomiques fines au sein des génocénoses même.

Il existe des exceptions à la cohérence géographique des génocénoses. Par exemple, la génocénose 2 (figure 4.11-A) est séparée à plusieurs endroits du globe. Ces stations se situent dans des conditions d'*upwelling* (remontée d'eau). Cela indique que de fortes variations des conditions environnementales influencent aussi la structure des génocénoses. De la même manière, il a été montré que la température, le nitrate, le phosphate, le fer et d'autres nutriments sont d'autres paramètres environnementaux qui diffèrent significativement entre les génocénoses. Néanmoins, les génocénoses ne sont pas les mêmes entre les différentes fractions de taille d'organismes (figure 4.11). Les différents types d'organismes ne sont donc pas affectés de la même manière par les différents facteurs environnementaux. Par exemple, les fractions de grande taille sont plus affectées par les variations de température que par la limitation en nutriments, contrairement aux petits organismes qui sont plus rapidement impactés par les variations en nutriments.

Enfin, la contribution principale de cette étude a été de montrer qu'à une échelle de temps inférieure à 1 an et demi, les courants océaniques sont le premier facteur à avoir une influence sur la dynamique de l'organisation génomique des communautés planctoniques. Pour cela, les distances de Simka ont été mises en corrélation avec le temps minimum que met le plancton à se déplacer entre chaque station. De plus, la corrélation entre ces deux facteurs dépend de la fraction de taille d'organismes considérée. Plus les organismes sont gros, moins leur diversité est impactée par les courants marins. Cela peut s'expliquer, entres autres, par le fait que les organismes plus grands ont également une espérance de vie plus grande et sont transportés sur de plus grandes distances. D'ailleurs, cette remarque explique un plus grand étalement spatial des génocénoses pour les grandes fractions de taille d'organismes (figure 4.11) qui parviennent donc à traverser de multiples régions biogéochimiques. L'ensemble des résultats de cette étude suggère que les génocénoses sont des entités dynamiques, influencées par les futurs changements environnementaux.

Cette étude comporte également des tests de validation de Simka. Par exemple, une expérimentation montre que la distance qualitative de Bray-Curtis de Simka est bien corrélée avec la même distance basée sur des OTUs (figure 4.12). Cela complète nos expériences de validation effectuées avec des distances taxonomiques. La composition en k-mers est un bon remplacement

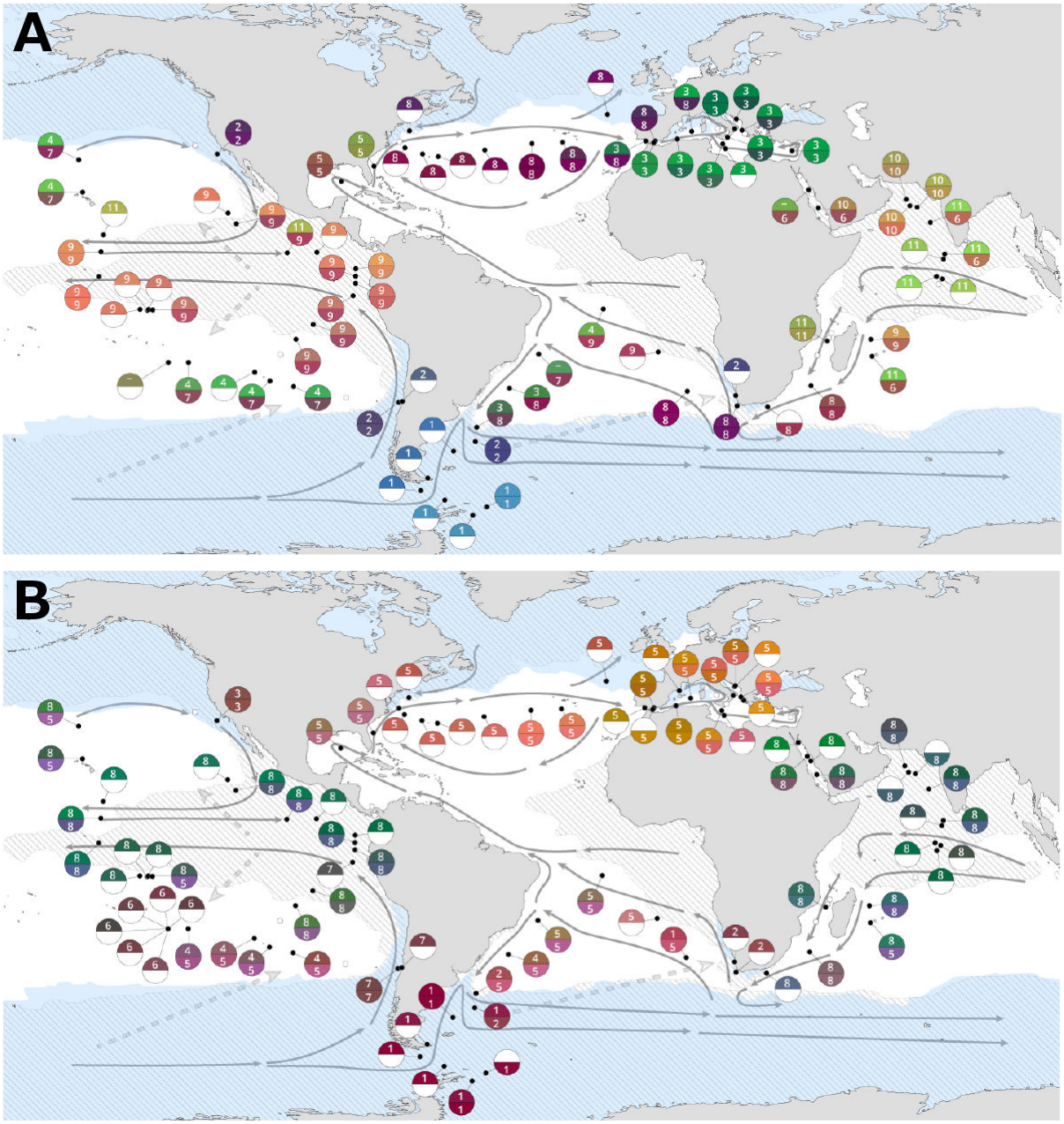

Figure 4.11. Représentation des stations de Tara Oceans et des génocénoses. Carte géographique des génocénoses pour la fraction d'organismes de (**A**) 0.8 à 5 µm (protistes) et (**B**) 180 à 2000 µm (métazoaires). Chaque cercle correspond à une station de Tara Oceans et contient le numéro de sa génocénose. Le numéro en haut du cercle correspond aux échantillons récoltés en surface de l'eau et le numéro en bas correspond aux échantillons récoltés à la profondeur où la concentration de chlorophylle est maximum. Les variations de couleur associées aux numéros indiquent des variations génomiques calculées par Simka.

à une estimation de la diversité même dans le cas de distances qualitatives. De plus, une étude taxonomique a montré que ce ne sont pas les mêmes organismes qui participent aux regroupements d'échantillons dans les génocénoses.

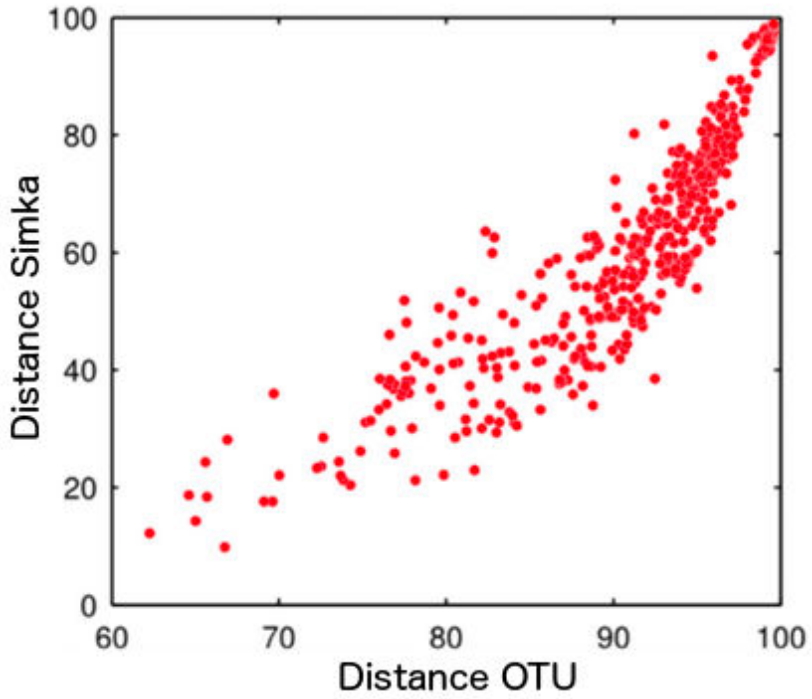

Figure 4.12. Corrélation entre des distances basées sur des OTUs et des distances de Simka. Chaque point représente une paire d'échantillons du projet Tara Oceans (fraction bactérienne). La coordonnée X indique la distance qualitative de Jaccard basée sur des OTUs, et la coordonnée Y la distance de Jaccard de Simka. Les distances basées sur des OTUs ont été déterminées à partir de jeux d'amplicons, alors que les distances de Simka ont été calculées à partir de données plein-génomes.

4.3 Conclusion

Un point clé notable de notre proposition pour comparer les communautés est d'utiliser le contenu en k-mers seulement. Cela peut biaiser nos distances par rapport à des distances qui seraient calculées à partir de comptages d'espèces. Le biais peut aller dans les deux sens : d'un coté, les régions génomiques partagées ou les transferts horizontaux entre les espèces vont sous-estimer nos distances. D'un autre coté, l'hétérogénéité de la taille des génomes et les variations de composition en k-mers le long des génomes vont surestimer les distances. Toutefois, les approches basées sur des compositions d'espèces ne sont pas applicables sur de grands jeux de données provenant d'environnements complexes (sol, eau de mer) à cause du manque de bonnes références. Notre proposition a l'avantage d'être une approche *de novo*, non biaisée par l'inconsistance et l'incomplétude des bases de données de références. De plus, nos expériences sur le projet HMP montrent que les distances quantitatives basées sur les k-mers sont bien corrélées avec les distances taxonomiques (corrélation de Spearman > 0.88 pour $k \geq 21$) et par conséquence que Simka retrouve les mêmes structures biologiques que les études taxonomiques. L'application sur les données de Tara Oceans a permis d'observer des regroupements d'échantillons cohérents en utilisant une distance qualitative de Simka. Une comparaison des résultats de Simka à des distances basées sur des OTUs démontre que la composition en k-mers est un bon remplacement à une estimation de la diversité. Cette application représente un autre grand pas vers la validation de notre proposition.

Deux intérêts nous ont motivés pour calculer une collection de distances plutôt qu'une seule comme la plupart des outils de l'état de l'art : différentes distances capturent différentes ca-

ractéristiques des données et toutes les distances calculées par SIMKA ont en commun le fait qu'elles sont additives sur les k-mers et peuvent donc être calculées simultanément en utilisant le même algorithme. En guise de support pour le premier point, nous avons vu que MASH se comporte mal quand on considère les échantillons du projet HMP par tissu puisque cet outil ne peut prendre en compte que des informations de présence-absence et non d'abondances relatives, contrairement à SIMKA. Les différences en abondances relatives sont des signaux subtils qui sont souvent au cœur d'intéressants résultats biologiques dans les études comparatives. Par exemple, Boutin *et al.* [138] ont montré que la structure entre différents échantillons de patients atteints d'une maladie pulmonaire était visible avec la distance quantitative de Bray-Curtis et absente avec la distance qualitative de Jaccard, mettant en évidence le rôle de l'abondances de certains microbes pathologiques dans la maladie. Dans d'autres études, il a été montré que la réponse des communautés bactériennes au stress ou à des changements environnementaux se caractérise par une augmentation de l'abondance de certains taxons rares [139, 140, 141, 142].

Le calcul des distance a une complexité en temps en $O(W \times N^2)$, où W est le nombre de k-mers distincts considérés et N est le nombre de jeux de lectures d'entrée. N est généralement limité à quelques centaines et ne peut pas être réduit. Cependant, W peut atteindre les centaines de milliards. Le filtre d'abondance fournit déjà un très grand gain de performances sans affecter les résultats des distances quantitatives, au moins sur les tests effectués sur les jeux de données du projet HMP. Cependant, ces données ne sont pas représentatives de tous les projets métagénomiques et, dans certains cas, ce filtre pourrait ne pas être désiré. Par exemple, dans le cas d'échantillons avec une très faible couverture ou quand une étude qualitative est effectuée dans laquelle les espèces rares ont plus d'impact. Dans ce cas là, il est important de noter que SIMKA peut tout de même passer à l'échelle sur de grands jeux de données même lorsque le filtre d'abondance est désactivé. Nous avons montré que SIMKA parvient à retrouver la principale structure des jeux de lectures peu couverts du projet GOS avec ou sans filtre d'abondance. Dans le chapitre suivant, nous testons d'autres approches que le filtre d'abondance pour réduire la quantité de k-mers à traiter et nous mesurons leurs effets sur les distances.

Chapitre 5

Approches de sous-échantillonnage de données pour le calcul des distances

Dans ce chapitre, nous testons et évaluons diverses approches de sous-échantillonnage de données. L'intérêt de telles approches est l'amélioration des performances globales de Simka. Le processus consiste à sélectionner aléatoirement un sous-échantillon à partir duquel est calculée une estimation des distances. La qualité est mesurée en comparant les résultats observés aux résultats attendus de Simka.

Plusieurs faits nous ont motivé à explorer des approches de sous-échantillonnage. L'outil Mash [108], présenté en section 2.3.1, a montré que l'index de Jaccard peut être estimé très précisément en utilisant seulement quelques milliers de k-mers. En plus d'une rapidité de calcul évidente, cette estimation est calculée par Mash avec une empreinte mémoire et une utilisation du disque négligeables. Cependant, Mash se limite au calcul de cette distance qualitative. Nous souhaitons voir si d'autres distances peuvent être estimées de la même manière afin de compléter les résultats de Mash. En particulier, nous avons montré dans le chapitre précédent que les indices quantitatifs sont vraiment importants. On peut imaginer que cette famille de distances est naturellement robuste au sous-échantillonnage. En effet, une distance quantitative est d'avantage impactée par les lectures provenant des espèces abondantes. Ces lectures abondantes ont plus de chance d'être présentes dans un échantillon aléatoire car elles sont plus nombreuses que les lectures provenant d'espèces rares.

La section 5.1 détaille le protocole de sous-échantillonnage. Ce protocole introduit des définitions, les données et les métriques d'évaluation des distances issues du sous-échantillonnage. Les trois sections suivantes évaluent trois approches de sous-échantillonnage distinctes. Dans la première, le sous-échantillonnage des jeux de données est effectué au niveau des lectures (section 5.2). Dans la deuxième, on évalue une approche de sous-échantillonnage au niveau des vecteurs d'abondances (section 5.3). Après avoir pris connaissances des qualités et des défauts de cette seconde approche, nous proposons et évaluons une nouvelle méthode d'estimation d'une distance de Simka (section 5.4).

5.1 Protocole de sous-échantillonnage et d'évaluation

Nous distinguons deux types de sous-échantillonnage : au **niveau des lectures** et au **niveau des *k*-mers**. Dans le premier cas, les distances de Simka sont calculées sur un sous-ensemble des lectures sélectionnées aléatoirement. Dans le deuxième cas, les jeux sont vus comme des ensembles de k-mers et seule une fraction d'entre eux est sélectionnée. Mash se situe dans la deuxième catégorie. Le sous-échantillonnage au niveau des lectures n'a jamais été exploré dans

la littérature.

Nous distinguons également deux types de distances : **quantitatives** et **qualitatives**. Comme nous l'avons mentionné, on peut penser qu'une distance quantitative est plus robuste au sous-échantillonnage. En revanche, les distances qualitatives sont impactées équitablement par tous les éléments. Dans ce cas, on peut imaginer que chaque lecture est importante. Nous avons choisi deux distances populaires pour représenter chaque type de distance : la distance quantitative de Bray-Curtis et la distance qualitative de Jaccard.

Données. Afin de s'assurer de la généralisation des résultats, nous avons considéré deux projets métagénomiques très différents. Le premier contient 100 jeux de données intestinaux du projet HMP ayant au moins 40 millions de lectures chacun. Le second contient 100 jeux de données océaniques du projet Tara Oceans contenant majoritairement des bactéries (filtre de taille d'organismes de 0.22 à 3 µm). Ceux-ci contiennent au moins 150 millions de lectures chacun.

Définitions. Une technique de sous-échantillonnage possède deux paramètres importants : **l'espace de sous-échantillonnage** et **la profondeur de sous-échantillonnage**. L'espace de sous-échantillonnage est l'ensemble des éléments qui peuvent être sélectionnés. La profondeur de sous-échantillonnage représente le nombre d'éléments sélectionnés aléatoirement dans un espace de sous-échantillonnage donné. Dans nos expériences, nous allons toujours considérer une sélection aléatoire uniforme sans remise des éléments. Chaque élément a donc la même probabilité d'être sélectionné et il ne peut pas être choisi plus d'une fois.

Pour évaluer l'impact du sous-échantillonnage, nous allons comparer les **résultats observés**, calculés à partir de sous-échantillons, aux **résultats attendues** de Simka. Les résultats attendus sont ceux qui considèrent tous les éléments de l'espace de sous-échantillonnage. Pour mesurer la variabilité des résultats due à la sélection aléatoire, plusieurs **réplicats** sont calculés à une profondeur donnée.

Métrique d'évaluation du sous-échantillonnage. Nous mesurons l'impact du sous-échantillonnage de 3 manières différentes :

— **Mesure de l'erreur relative.** Pour une paire d'échantillons donnée, cette métrique mesure précisément l'erreur d'estimation de leur distance par rapport à la valeur attendue. Pour que cette mesure soit plus intuitive à interpréter, nous représentons les distances comme des similarités (similarité = 1 - distance). La mesure de l'erreur relative est donnée par la formule suivante :

$$ErreurRelative = 100 \times \frac{S_{obs} - S_{att}}{S_{att}}$$

où S_{obs} et S_{att} sont respectivement une similarité observée et une similarité attendue. Cette métrique indique de combien de pourcentage l'estimation est éloignée de la similarité attendue. Pour un projet et une profondeur de sous-échantillonnage donnés, nous obtenons une série de $N \times N$ erreurs d'estimation où N est le nombre de jeux de données du projet.

— **Corrélation entre les résultats attendus et observés.** Après avoir mesuré l'erreur d'estimation, nous considérons les distances d'une manière relative. En une valeur, nous résumons l'erreur sur toute la matrice de distances. Nous cherchons à savoir si l'erreur d'estimation impacte l'ordre des paires d'échantillons triées par similarité. Pour cela, la corrélation entre les résultats attendus et observés est mesurée quantitativement grâce à la corrélation de Spearman et qualitativement en observant leur nuage de points.

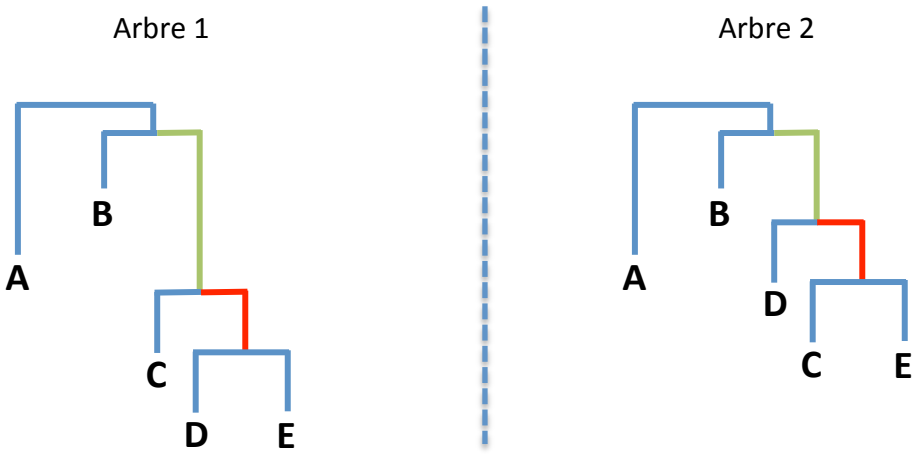

FIGURE 5.1. Concept de bipartition dans un clustering hiérarchique. Les branches vertes séparent les jeux de données en une même bipartition dans les deux arbres : (A, B) et (C, D, E). Cette bipartition est donc partagée entre les arbres. Les branches rouges séparent les jeux de l'arbre 1 en (A, B, C) et (D, E), et ceux de l'arbre 2 en (A, B, D) et (C, E). Elles représentent deux bipartitions dissimilaires. Il y a autant de bipartitions que de branches internes (c'est-à-dire toutes les banches sauf les feuilles).

— **Comparaison de leur classification.** Le calcul des matrices de distances de SIMKA est classiquement suivi d'un clustering des jeux de données. Nous proposons de mesurer les différences entre le clustering attendu et celui observé. Pour classifier les jeux de données, nous avons utilisé UPGMA [134], une technique classique de clustering hiérarchique. Le concept de **bipartition** a été utilisé pour comparer deux arbres hiérarchiques d'un point de vue topologique. Une bipartition est la séparation d'un arbre en deux parties connectées par une seule branche. Les jeux de données se retrouvent séparés en deux groupes, mais leurs relations à l'intérieur des groupes ne sont plus considérées (figure 5.1). Une bipartition est dite partagée si elle est présente dans les deux arbres, ou dissimilaire si elle apparait dans un arbre mais pas dans l'autre. La distance topologique entre deux arbres C_1 et C_2 est le nombre de bipartitions dissimilaires parmi l'ensemble des bipartitions [143]. Nous avons normalisé ce nombre par le nombre total de branches internes pour une meilleure interprétation :

$$DistanceTopologique(C_1, C_2) = 100 \times \frac{bipartitions\ dissimilaires}{biparitions\ totales}$$

Pour des raisons pratiques, nous n'avons pas considéré de métriques prenant en compte la longueur des branches [144]. Les implémentations disponibles ne normalisent pas leurs résultats par la longueur de branche totale. Les résultats sont donc difficilement comparables et interprétables.

Nos travaux sur le sous-échantillonnage au niveau des lectures, présentés dans la section suivante, sont très préliminaires. Nous avons pu évaluer la robustesse des deux types de distances au sous-échantillonnage. Cependant, nous ne sommes pas en mesure de lier le nombre de lectures utilisées à l'erreur d'estimation de la distance. Nos travaux au niveau des k-mers, présentés ensuite, sont quant à eux assez avancés. Ils ont abouti à un nouvel outil permettant d'estimer la distance de Bray-Curtis de SIMKA avec une grande précision en utilisant peu de k-mers.

5.2 Sous-échantillonnage au niveau des lectures

Dans cette section, nous évaluons l'impact du sous-échantillonnage au niveau des lectures sur la distance Bray-Curtis et sur la distance de Jaccard de Simka. Nous avons choisi un même espace de sous-échantillonnage de 40 millions de lectures pour les projets intestinal et océanique afin de pouvoir comparer leurs résultats. Neufs profondeurs de sous-échantillonnage ont été considérées : de 10 à 90% de l'espace par pas de 10%. Pour chaque profondeur, 10 réplicats ont été calculés pour évaluer l'impact de la sélection aléatoire.

5.2.1 Erreur d'estimation des distances issues du sous-échantillonnage

Les erreurs d'estimation de toutes les matrices de distances de Bray-Curtis et de Jaccard (100 × 100 distances) sont tout d'abord mesurées en utilisant un seul réplicat (figures 5.2-A-B). Sur le projet intestinal, la distance de Bray-Curtis atteint une erreur acceptable à partir de 30% de lectures sélectionnées (90% des erreurs sont comprises entre -2% et -10%). À la même profondeur de sous-échantillonnage, l'erreur est plus élevée sur le projet océanique (médiane = -23%), mais les erreurs varies dans une plage restreinte (entre -17% et -29%). La distance de Jaccard converge beaucoup plus difficilement vers les valeurs attendues. Son erreur est acceptable sur les deux projets à partir de 70% de lectures utilisées (90% des erreurs comprises entre -2% et 8% pour le projet intestinal et entre -4% et -12% pour le projet océanique). À l'inverse des distances de Bray-Curtis, les distances de Jaccard du projet intestinal sont moins bien estimées que celles du projet océanique. Cette première expérience montre qu'à nombre de lectures égales, la précision des estimations est très hétérogène selon le type de distances et projets considérés.

Pour évaluer l'impact de la sélection aléatoire, nous observons la distance d'une paire d'échantillons dans 10 réplicats. Nous avons choisi une distance proche de la médiane de la matrice de distance. Les figures 5.2-C-D montrent que la sélection aléatoire n'a aucun impact quel que soit le type de distances et le projet considérés. Les distances issues du sous-échantillonnage convergent vers la même valeur. Cependant, cette valeur n'est pas la distance attendue de Simka.

Aucune des distances ne convergent rapidement vers les distances attendues. Cependant, à partir d'un certain nombre de lectures utilisées, les plages d'erreurs effectuées sont restreintes. On peut alors espérer que les estimations auront peu d'impact sur l'ordre attendu des paires d'échantillons par similarité.

5.2.2 Corrélation entre les distances attendues et les distances observées

Malgré l'erreur d'estimation effectuée, les distances de Bray-Curtis observées sont extrêmement bien corrélées aux distances attendues (figures 5.3 gauche). Le coefficient de corrélation de Spearman est supérieur à 0.99 pour chaque projet, même à 10% de lectures considérées. Cette conclusion vaut aussi pour la distance de Jaccard sauf dans le cas du projet intestinal (figures 5.3 droite). Nous pouvons espérer que les corrélations élevées se reflètent sur une bonne classification des échantillons.

5.2.3 Impact sur la classification des échantillons

Les figures 5.4 et 5.5 montrent respectivement les classifications du projet intestinal et du projet océanique obtenues par Simka en considérant toutes les lectures (40 millions par jeu de données). Les arbres ont été annotés pour indiquer les bipartitions qui n'apparaissent pas dans les classifications obtenues à partir de 30% des lectures seulement. On peut remarquer que la distance de Bray-Curtis obtient des clustering moins bruités (branches internes plus longues)

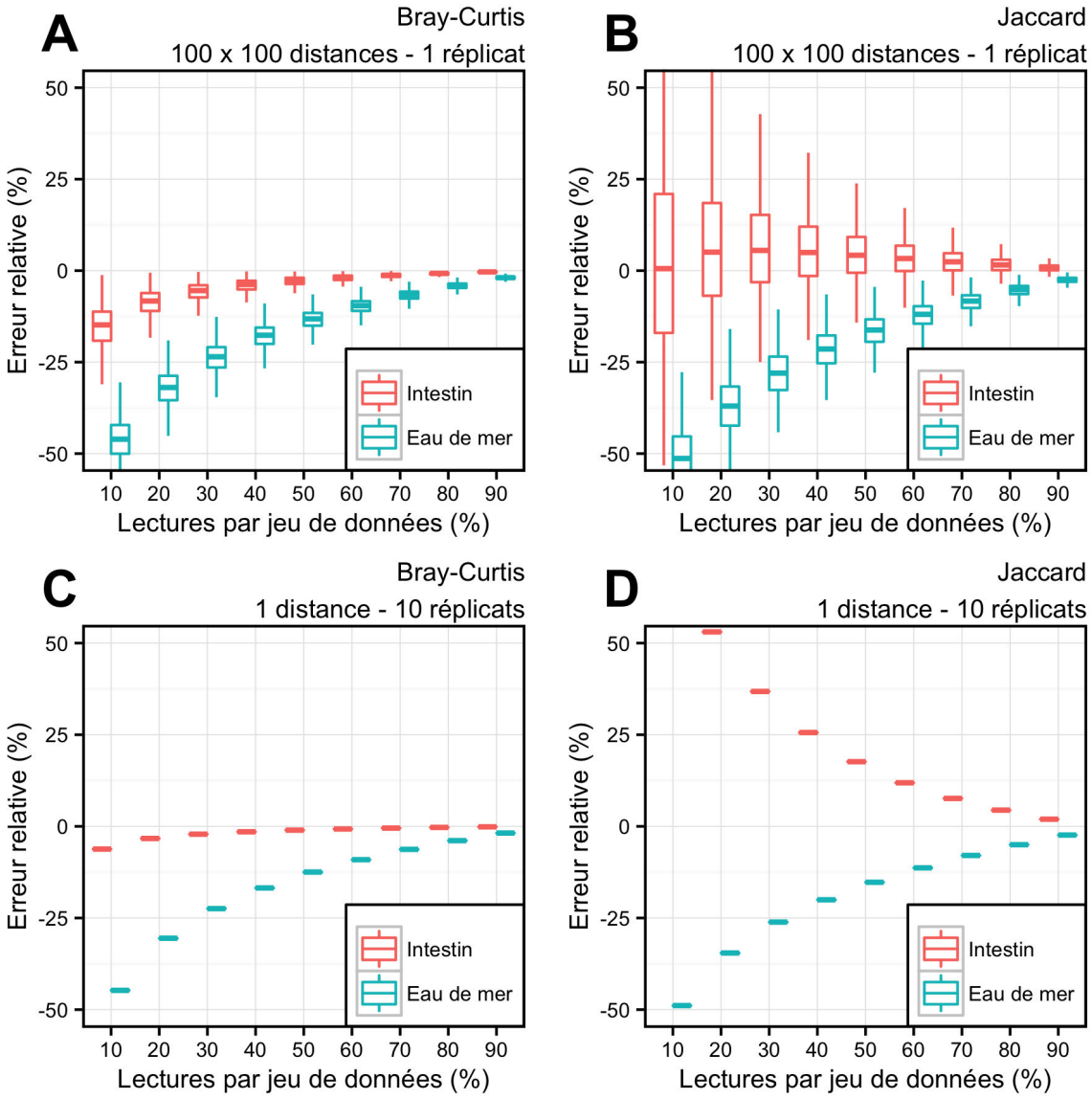

Figure 5.2. Erreur d'estimation des distances issues du sous-échantillonnage au niveau des lectures. Les lectures ont été sélectionnées aléatoirement dans un espace de 40 millions de séquences. (**A-B**) Erreurs d'estimation de toutes les distances par projet (100 × 100 distances) en utilisant un seul réplicat. (**C-D**) Erreurs d'estimation d'une seule distance dans 10 réplicats (les boîtes à moustaches ne sont pas visibles car il n'y a pas de variabilité).

que la distance de Jaccard. Ces branches plus longues sont plus robustes vis à vis du sous-échantillonnage. Par conséquent, la distance de Bray-Curtis retrouve mieux le clustering attendu. Ces figures montrent également que la classification du projet océanique est mieux retrouvée que celle du projet intestinal. Cet effet est entièrement lié aux données. Les échantillons d'eau de mer utilisés sont naturellement très structurés car ils proviennent d'océans et de zones biochimiques bien distinctes. À l'inverse, les échantillons intestinaux proviennent d'un même tissu d'individus en bonne santé et sont donc plus difficilement dissociables. On retrouve tout de même de petites structures locales bien conservées de 2 à 5 échantillons intestinaux.

Nous avons quantifié les différences topologiques à plusieurs profondeurs de sous-échantillonnage. Nous jugeons que la classification est fiable si le nombre de bipartitons dis-

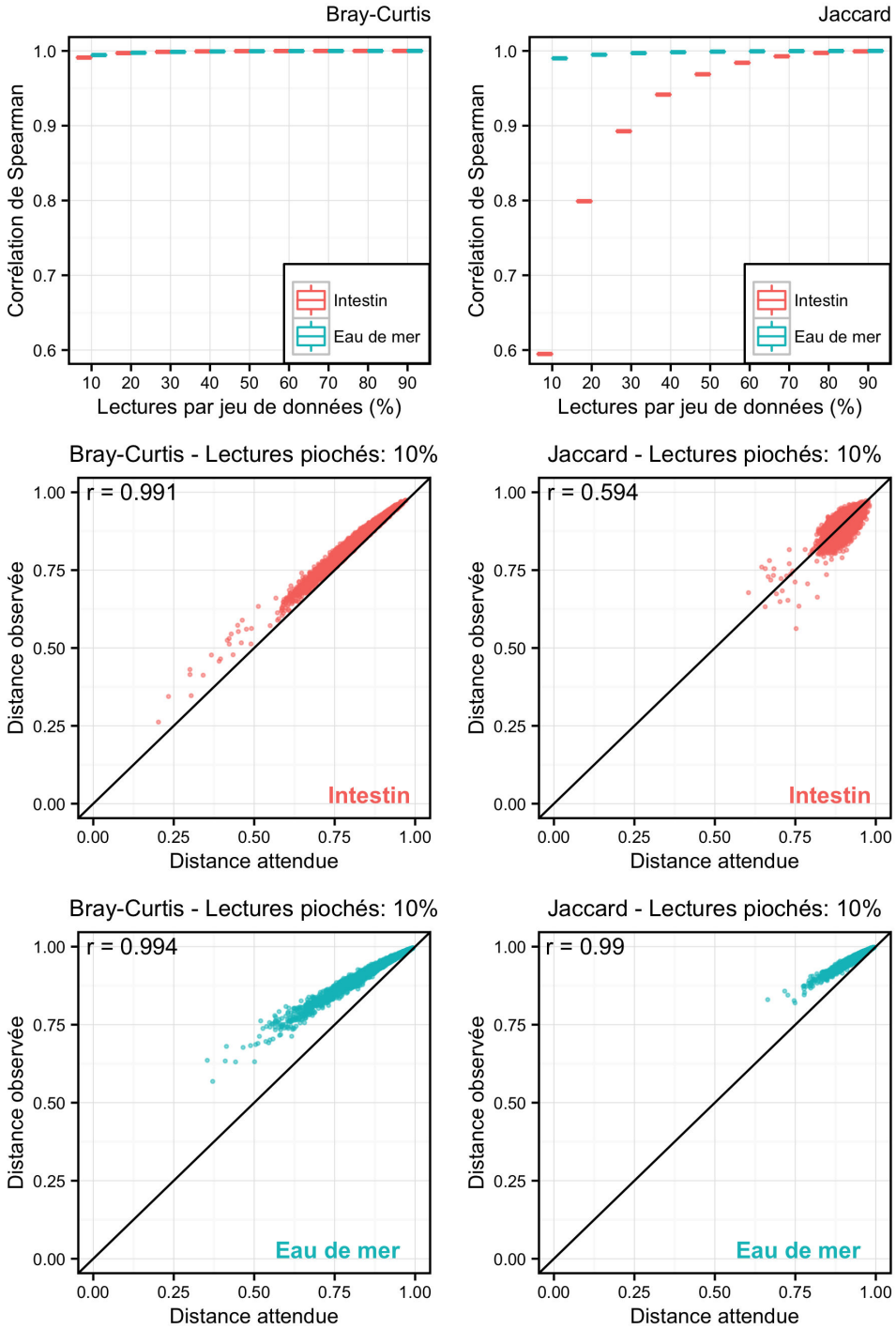

Figure 5.3. Corrélation entre les distances de Simka et les distances issues du sous-échantillonnage au niveau des lectures. Dans chaque test, dix matrices de distances (réplicats) de taille 100 × 100 sont comparées à la matrice attendue de Simka. Les nuages de points montrent les distances attendues de Simka contre les distances observées à 10% de lectures sélectionnées par jeu de données.

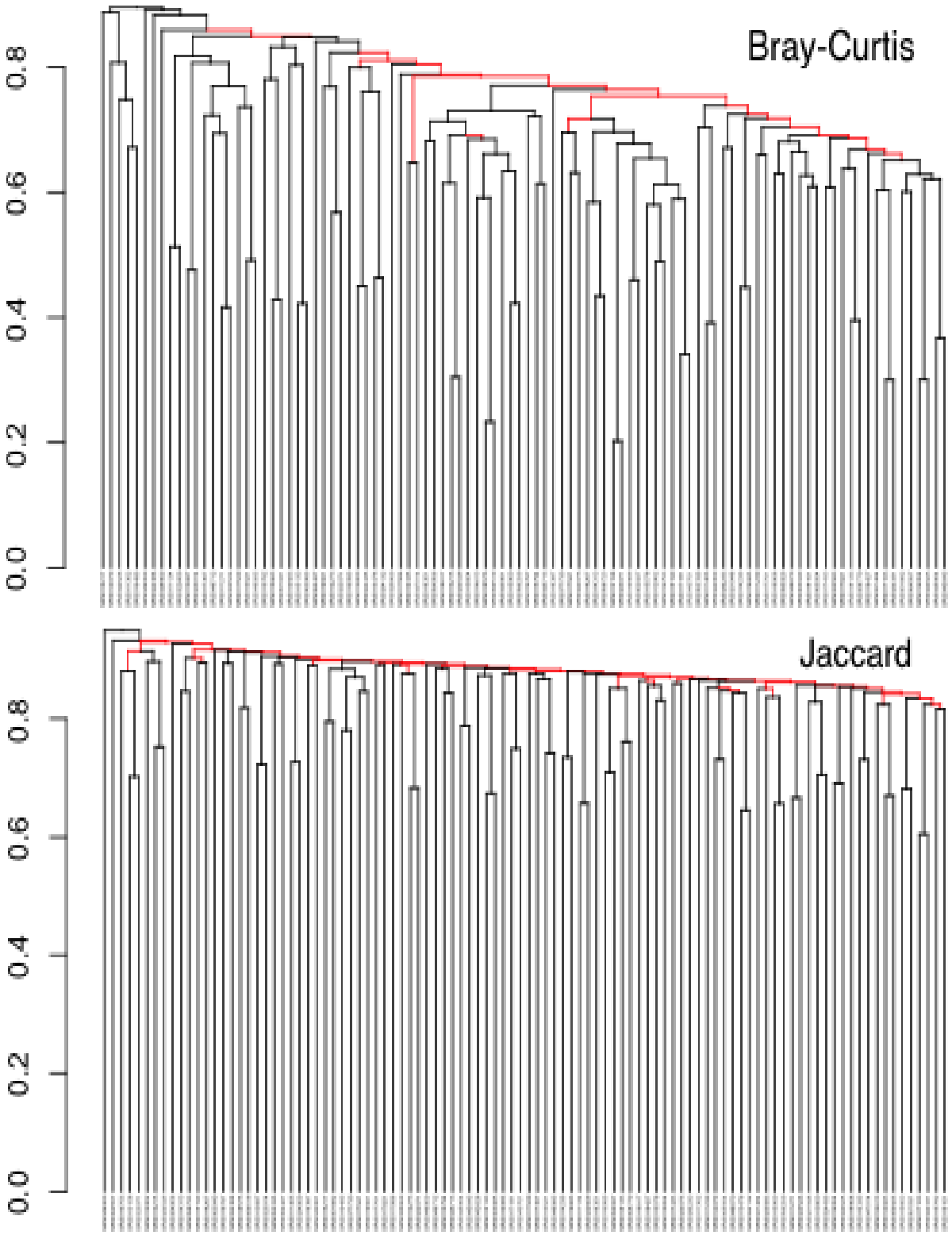

Figure 5.4. Impact du sous-échantillonnage au niveau des lectures sur la classification des jeux de données intestinaux. Classifications des jeux de données intestinaux calculée à partir de toutes les lectures. Une branche rouge indique que ses bipartitions n'apparaissent pas dans une classification obtenue en n'utilisant que 30% des lectures.

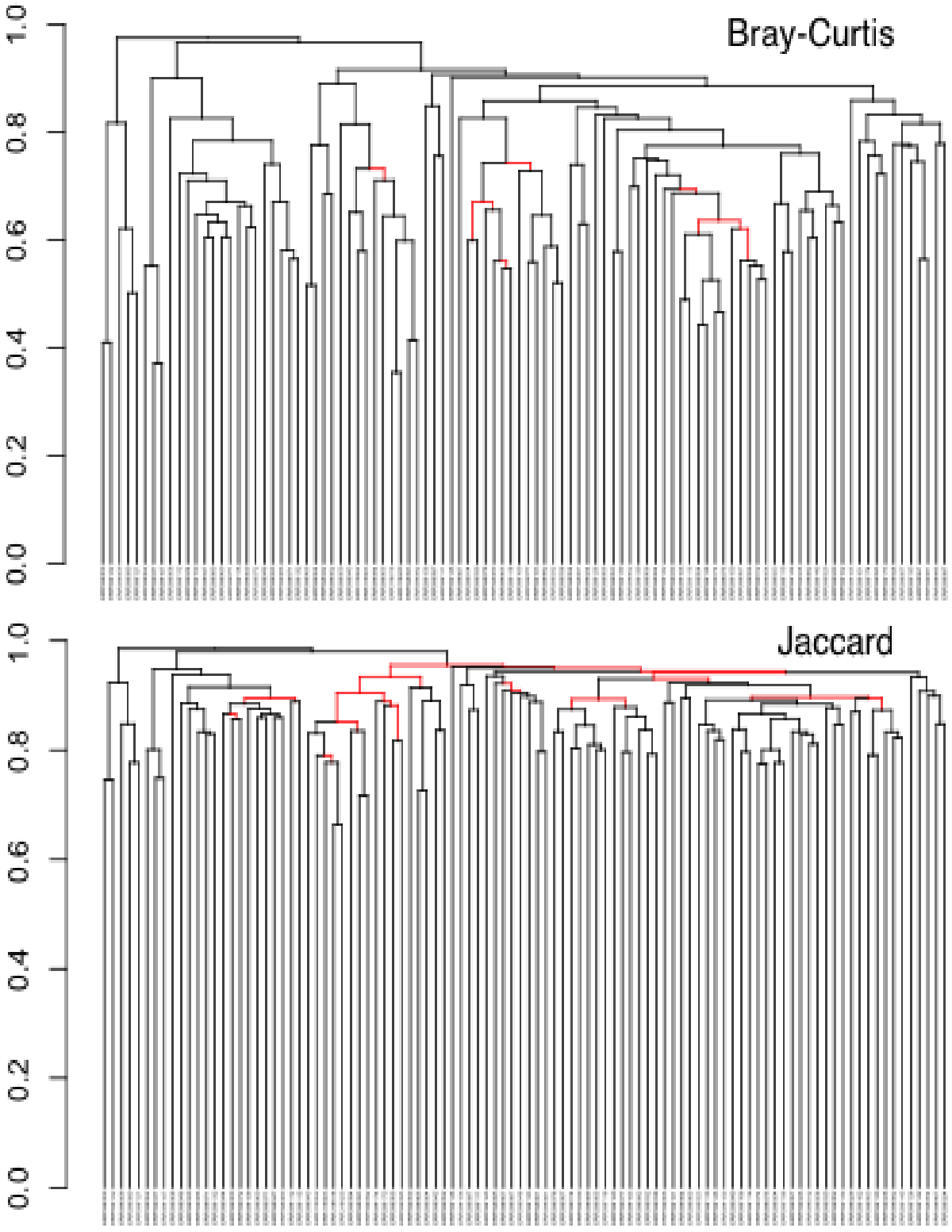

Figure 5.5. Impact du sous-échantillonnage au niveau des lectures sur la classification des jeux de données océaniques. Classifications des jeux de données océaniques calculée à partir de toutes les lectures. Une branche rouge indique que ses bipartitions n'apparaissent pas dans une classification obtenue en n'utilisant que 30% des lectures.

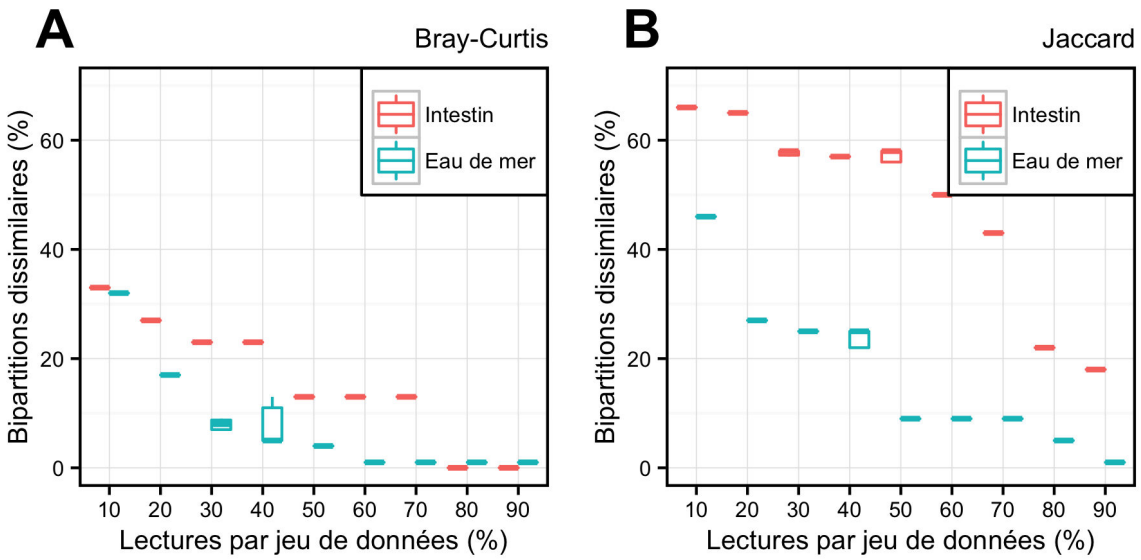

Figure 5.6. Impact du sous-échantillonnage au niveau des lectures sur la classification des jeux de données. Les jeux de données intestinaux et océaniques ont été regroupés hiérarchiquement avec la méthode UPGMA. Les figures (**A**) et (**B**) indiquent le pourcentage de bipartitions dissimilaires entre le clustering attendu et observé pour la distance de Bray-Curtis et de Jaccard. Dix réplicats ont été considérés par test.

similaires est inférieur à 10%. La distance de Bray-Curtis parvient à retrouver la structure topologique du projet océanique à partir de 30% de lectures considérées et celle du projet intestinal à partir de 80% de lectures (figure 5.6-A). La distance de Jaccard ne retrouve jamais une classification fiable des jeux intestinaux. Elle retrouve celle du projet océanique en utilisant 50% de lectures (figure 5.6-B). Ces différences entre la distance de Jaccard et la distance de Bray-Curtis montrent encore une fois que cette dernière est beaucoup plus robuste au sous-échantillonnage.

5.2.4 Conclusion

La précision des estimations varie en fonction du type de distances et des projets considérés. Dans des tests personnels, nous avons également vu que les erreurs ne sont pas les mêmes si la taille de l'espace de sous-échantillonnage change. Il semble donc difficile de prédire l'erreur effectuée en fonction du nombre de lectures utilisées. Cette prédiction aurait permis à un utilisateur de faire un compromis entre qualité des résultats et performances des calculs. Elle pourrait également orienter un utilisateur sur l'effort de séquençage à produire si ses données n'étaient pas encore séquencées. Quoi qu'il en soit, cette étude montre que les distances ne sont pas robustes à un sous-échantillonnage excessif au niveau des lectures. La distance de Bray-Curtis autorise une marge de sous-échantillonnage où la classification des jeux de données reste la même (environ 80% des données). Cette propriété est importante puisqu'il faut rappeler que les jeux de données réelles ne sont eux mêmes que des échantillons de la communauté présente. À l'inverse, l'estimation d'une distance qualitative se dégrade rapidement en fonction du nombre de lectures considérées. Les conclusions tirées à partir de ce type de distances sont plus sensibles à l'effort de séquençage produit.

5.3 Sous-échantillonnage au niveau des vecteurs d'abondances

Le sous-échantillonnage est maintenant effectué au niveau des k-mers sur un ensemble fixe de lectures. La seconde approche de sous-échantillonnage que nous avons testée ne considère qu'une fraction des vecteurs d'abondances disponibles. Pour rappel, après avoir compté les k-mers de chaque jeu de données, SIMKA fusionne les spectres et génère les vecteurs d'abondances. Un vecteur d'abondances est un k-mer distinct et ses N comptages dans les N jeux de données.

Nous avons calculé l'erreur de cette estimation par rapport aux résultats attendus de SIMKA. Les distances de SIMKA considèrent des milliards de vecteurs d'abondances : 15 milliards pour le projet intestinal et 350 milliards pour le projet océanique. Les distances issues du sous-échantillonnage utilisent un nombre fixe de vecteurs d'abondances allant de 100 à 10 millions. Ils sont sélectionnés aléatoirement sans remise. Nous n'évaluons que la distance de Bray-Curtis puisque l'outil MASH calcule déjà la distance de Jaccard au niveau des k-mers très efficacement.

5.3.1 Erreur d'estimation des distances issues du sous-échantillonnage

La figure 5.7 montre l'erreur d'estimation par rapport au nombre de vecteurs d'abondances utilisés. Sur le projet intestinal, une valeur d'estimation satisfaisante est très rapidement atteinte : 90% des erreurs sont comprises entre -7% et 4% à 1 million de vecteurs utilisés. Cela ne représente que 0.007% de l'espace de sous-échantillonnage. L'estimation est moins bonne sur le projet océanique. Cette erreur reste acceptable compte tenu de la faible quantité de données considérées (90% des erreurs comprises entre -5% et -20% en utilisant 0.003% des vecteurs). Les nuages de points confirment que les distances observées convergent rapidement vers les distances attendues (figure 5.7).

5.3.2 Conclusions

Cette expérience montre que la distance de Bray-Curtis peut être estimée avec très peu de k-mers, tout comme l'index de Jaccard de MASH. Cependant, cette approche à deux faiblesses majeures. Premièrement, la sélection au niveau des vecteurs d'abondances ne garantit pas que le nombre de k-mers distincts utilisés est le même pour chaque paire de jeux. Les jeux ayant plus de k-mers distincts génèrent d'avantage de vecteurs d'abondances. En particulier, nous avons remarqué que les jeux de données océaniques ont des nombres de k-mers distincts très variables (de 1 à 6 milliards), alors que les jeux de flore intestinale saturent à un nombre de k-mers distincts similaires. Cela peut expliquer les différences d'erreurs d'estimation entre les deux projets. Deuxièmement, le gain en performances est moindre. Cette approche n'améliore que le temps de calcul de la distance de Bray-Curtis. Or, c'est déjà l'étape la plus rapide dans le processus de SIMKA (figure 3.5). Le comptage des k-mers et la fusion des spectres ont toujours besoin d'être effectués intégralement. Dans la section suivante, nous proposons une nouvelle méthode de sous-échantillonnage répondant à ces deux problèmes.

5.4 SimkaMin : nouvelle méthode d'estimation de la distance de Bray-Curtis

Cette section présente tout d'abord notre nouvelle méthode de sous-échantillonnage au niveau des k-mers, nommée SIMKAMIN. SIMKAMIN a pour objectif d'estimer la distance de Bray-Curtis de SIMKA. La qualité de l'estimation est ensuite évaluée. Enfin, nous comparons les performances de SIMKAMIN à ceux de SIMKA et de MASH en termes de temps de calcul.

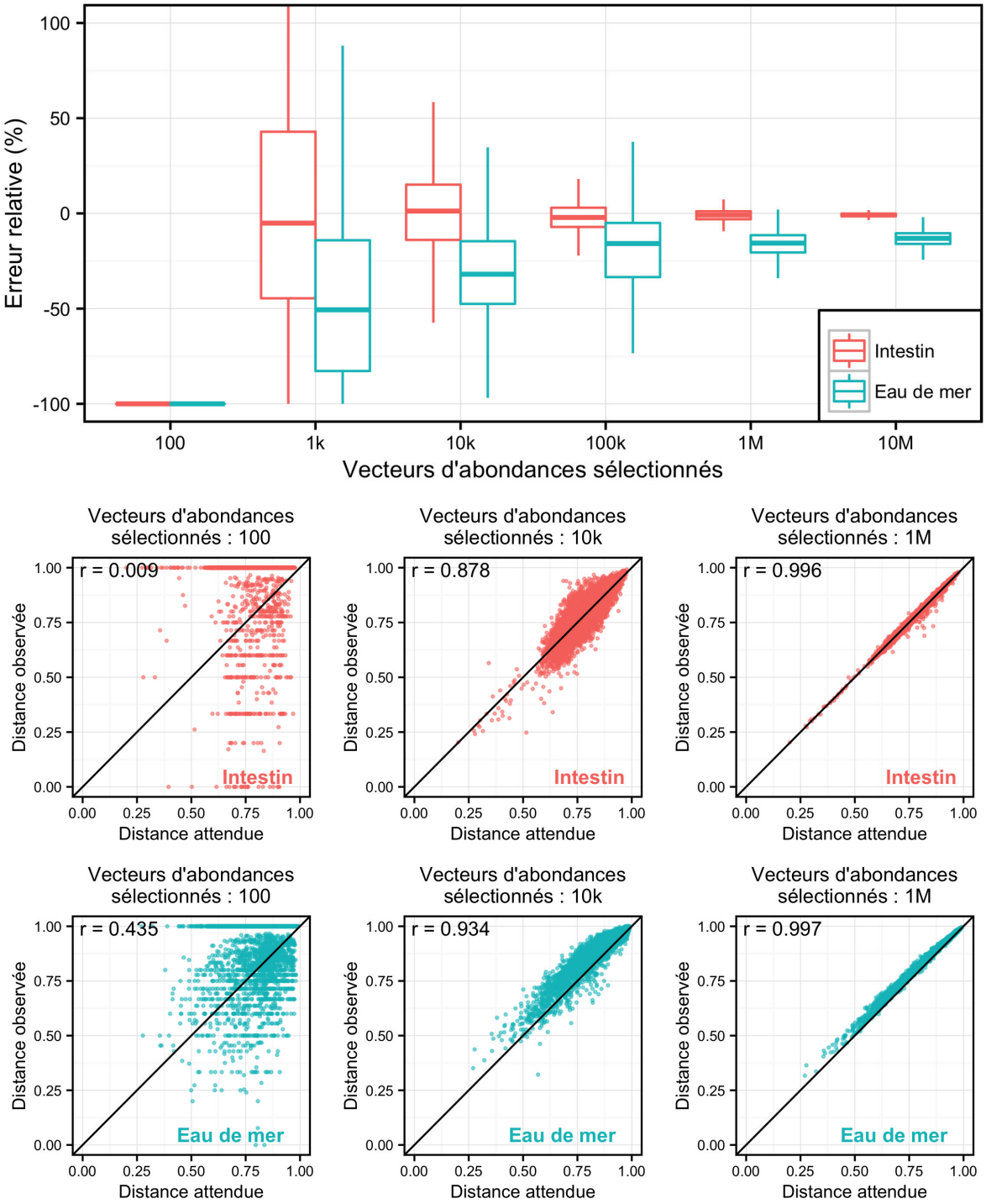

Figure 5.7. Impact du sous-échantillonnage au niveau des vecteurs d'abondances sur la distance de Bray-Curtis. Dans chaque test, 100 réplicats constitués de 100 × 100 distances sont considérés pour chaque projet. L'erreur est calculée par rapport aux distances attendues de Simka qui a utilisé plusieurs milliards de vecteurs d'abondances. Les nuages de points montrent les distances attendues de Simka contre celles issues du sous-échantillonnage à trois profondeurs de sous-échantillonnage.

5.4.1 Méthode

Le fonctionnement théorique de SimkaMin est le suivant. Pour chaque paire de jeux de données :

1. Fusionner leur spectre de k-mers.
2. Sélectionner aléatoirement n vecteurs d'abondances (de taille 2).
3. Calculer la distance de Bray-Curtis basée sur les n vecteurs sélectionnés.

Cette méthodologie résout le problème de sélection non uniforme de l'approche présentée précédemment. Pour améliorer les performances globales, nous utilisons la méthode Minhash utilisée par l'outil Mash, présentée en section 2.3.1. La méthode Minhash répond parfaitement bien à notre problème. Elle sélectionne aléatoirement des k-mers distincts au niveau du spectre conjoint de deux jeux de données sans avoir besoin de les fusionner explicitement. Cependant, elle est initialement conçue pour calculer l'index de Jaccard et ne considère donc pas l'abondance des k-mers. Notre méthode adapte Minhash pour obtenir le comptage exact des k-mers qu'elle sélectionne sans impacter le temps de calcul. La stratégie globale est fortement inspirée de l'outil Mash.

5.4.1.1 Sélection des k-mers.

Pour rappel, Mash utilise une fonction de hachage pour transformer les k-mers d'un jeu de données et les trier dans un ordre aléatoire. La *signature* d'un jeu de données correspond aux n plus petits éléments au sens de la fonction de hachage choisit. L'algorithme 2 détaille notre nouveau processus de sélection. La différence avec l'algorithme de Mash est l'utilisation d'un dictionnaire, noté KC, pour associer des k-mers à leur abondance (ligne 1). La liste L est triée par ordre croissant à la volée dès qu'on y insère un entier (ligne 2). Elle contient au maximum n éléments. Son plus grand élément est noté L_{top} (ligne 3).

Chaque k-mer est lu (lignes 4-5), puis haché (ligne 6). La fonction de hachage transforme uniformément les k-mers dans un ensemble de très grands entiers. La probabilité de collision de deux k-mers distincts est très proche de zéro. La valeur de hachage du k-mer courant est noté H. Les lignes 7 à 12 sont seulement appelées tant que la liste L ne contient pas n éléments distincts. Avant d'insérer un élément dans L, on vérifie s'il est présent dans L en requêtant le dictionnaire KC (ligne 8). S'il n'est pas présent, on insère H dans la liste L et dans le dictionnaire KC (lignes 10 à 12). Le comptage de H est initialisé à 1 car c'est la première fois qu'il est vu. Si H est déjà présent dans KC, on incrémente son comptage (ligne 9). Une fois la liste L remplie, seules les valeurs de hachage plus petites que L_{top} sont considérées (ligne 14). La suite de l'algorithme a un fonctionnement similaire à sa première partie. Si un élément est plus petit que L_{top} et qu'il est vu pour la première fois (il n'est pas dans le dictionnaire KC), alors la valeur L_{top} est retirée de KC et de la liste L et on y insère l'élément courant. La liste L contient donc à tout moment les n plus petits k-mers hachés. Le dictionnaire KC maintient leur abondance. La *signature* d'un jeu de données est maintenant l'ensemble des n plus petits k-mers hachés et leur comptage.

5.4.1.2 Calcul de la distance de Bray-Curtis.

Une fois les signatures de chaque jeu de données calculées, elles sont comparées par paire de la même manière que dans l'algorithme Minhash (figure 2.3). Deux signatures sont fusionnées pour calculer efficacement les termes de la distance de Bray-Curtis. Il est important de rappeler que la fusion s'arrête lorsque n éléments distincts ont été vus. De cette manière, toutes les paires de jeux utilisent exactement n k-mers distincts. Notons que la distance qualitative de Jaccard (calculée par Mash) peut être calculée en même temps en ignorant le comptage des k-mers.

Entrées : Un jeu de lectures S, le nombre de k-mers distincts à selectionner n.
Output : Les n k-mers distincts sélectionnés et leur abondance exacte dans S.

```
KC ← dictionnaire de comptage de k-mers vide
L ← liste vide (triée à chaque insertion d'un élément)
L_top : plus grande valeur insérée dans L à tout moment
pour chaque Séquence s dans S faire
    pour chaque k-mer K dans s faire
        H = hach(K)
        si taille de L < n alors
            si H dans KC alors
                Ajouter 1 au comptage de H dans KC
            sinon
                Insérer H dans L
                Insérer H dans KC avec comptage initilisé à 1
        sinon
            si H < L_top alors
                si H dans KC alors
                    Ajouter 1 au comptage de H dans KC
                sinon
                    Enlever L_top de L
                    Enlever L_top de KC
                    Insérer H dans L
                    Insérer H dans KC avec comptage initilisé à 1
retourner KC
```

Algorithme 2 : Sélection aléatoire de n k-mers distincts et leur abondance exacte dans un jeu de données. La liste L est automatiquement triée dés qu'un élément y est inséré (en $O(log\, n)$ opérations). La valeur L_{top} est automatiquement mise à jour en lisant le dernier élément de L.

5.4.1.3 Filtrage efficace des *k*-mers vus une seule fois.

La méthode présentée est facilement adaptable pour que la sélection soit effectuée au niveau des k-mers solides (ceux vus plus d'une fois dans un jeu). Cette technique a été introduite dans la publication de Mash. Pour pouvoir être sélectionné, un k-mer doit être vu au moins deux fois. Pour tester cela, une table de hachage de k-mers est utilisée, nommée $L_{candidats}$. Pendant le processus de sélection, si un k-mer est présent dans $L_{candidats}$, cela signifie qu'il a au moins été vu une fois auparavant. S'il n'y est pas, alors il est inséré dans $L_{candidats}$. Cette technique est triviale mais $L_{candidats}$ explose en mémoire sur les jeux de données considérés. Deux optimisations sont apportées pour réduire drastiquement l'empreinte mémoire de cette technique.

La première consiste à utiliser une structure de données plus compacte qu'une table de hachage standard, tel que le filtre de Bloom. Le filtre de Bloom a été introduit pendant la présentation de l'outil Compareads (section 2.2.2). Il s'agit d'une table de hachage qui ne gère pas les collisions au profit d'un énorme gain de mémoire. Dans notre application, les faux-positifs du filtre de Bloom impliquent que des k-mers vus une seule fois pourront être sélectionnés avec une faible probabilité ($< 0.1\%$).

La deuxième optimisation tire parti du fonctionnement de MINHASH. Si la valeur de hachage d'un k-mer est plus grande que L_{top}, alors celui-ci ne sera jamais considéré quel que soit son abondance (algorithme 2 - ligne 14). Il n'y a donc pas besoin d'insérer ces k-mers dans $L_{candidats}$. Ce filtre est extrêmement efficace. Plus le nombre de k-mers traités augmente, plus L_{top} devient petit. Par conséquent, de plus en plus faible est la probabilité qu'un k-mer passe le filtre. Ainsi, en pratique, il suffit d'un filtre de Bloom d'une centaine de Mo pour effectuer la sélection au niveau des k-mers distincts solides.

5.4.2 Évaluation de la distance de Bray-Curtis calculée par SimkaMin

Nous mesurons tout d'abord l'erreur d'estimation des distances de SIMKAMIN sur les jeux de données intestinaux et océaniques. Les distances attendues de SIMKA considèrent 500 millions de k-mers distincts par jeu en moyenne pour le projet intestinal et 3 milliards pour le projet océanique. Pour mesurer l'impact de la sélection aléatoire, nous avons calculé 10 réplicats en changeant la graine de la fonction de hachage. La précision de l'estimation de la distance de Bray-Curtis est ensuite comparée à celle de la distance de Jaccard calculée par MASH. Enfin, nous vérifions si le filtre d'abondance a un impact sur l'estimation.

5.4.2.1 Erreur d'estimation des distances de SimkaMin

L'erreur d'estimation des distances de SIMKAMIN a été mesurée par rapport aux résultats attendus de SIMKA. La figure 5.8 montre l'erreur observée en fonction d'un nombre croissant de k-mers distincts sélectionnés n. Les distances sont globalement moins bien estimées sur le projet océanique. Les résultats sont fiables à partir de $n = 100k$ (figure 5.8-A, 90% des erreurs comprises entre -8% et 8%). Cependant, il existe une forte variabilité due à la sélection aléatoire (figure 5.8-B - $n = 100k$ - erreurs comprises entre -3% et 21%). À un million de k-mers distincts sélectionnés, il n'existe quasiment plus d'erreur d'estimation pour les deux projets considérés. Cela représente seulement 0.2% des k-mers distincts par jeu de données du projet intestinal et 0.03% par jeu du projet océanique. La différence d'erreur d'estimation entre ces deux projets pourrait provenir de la différence de leur espace de sous-échantillonnage (6 fois plus grand dans le cas du projet océanique). Cela n'a pas été vérifié explicitement.

5.4.2.2 Comparaison avec l'estimation de la distance de Jaccard de MASH (minhash)

La précision de l'estimation de la distance de Bray-Curtis de SIMKAMIN a été comparée à celle de la distance de Jaccard fournie par MASH. La figure 5.9 montre l'erreur d'estimation sur un nombre croissant de k-mers distincts sélectionnés. À profondeur de sous-échantillonnage égale, la distance de Jaccard est toujours mieux estimée que la distance de Bray-Curtis. Il faut utiliser environ 10 fois plus de k-mers distincts pour estimer de manière identique la distance de Bray-Curtis et la distance de Jaccard (figure 5.9-A-B, n=100k et n=1M, par exemple). Les conclusions sont les mêmes pour les deux projets considérés.

5.4.2.3 Impact du filtre d'abondance

Nous avons comparé la précision de l'estimation de la distance de Bray-Curtis de SIMKAMIN en activant ou non le filtre d'abondance. Pour rappel, lorsque le filtre d'abondance est activé, seuls les k-mers vus au moins deux fois dans un jeu de données sont considérés. Les deux types de distances ont été comparés aux résultats filtrés et non filtrés de SIMKA. La figure 5.10 montre l'erreur d'estimation en fonction d'un nombre croissant de k-mers distincts sélectionnés. On peut

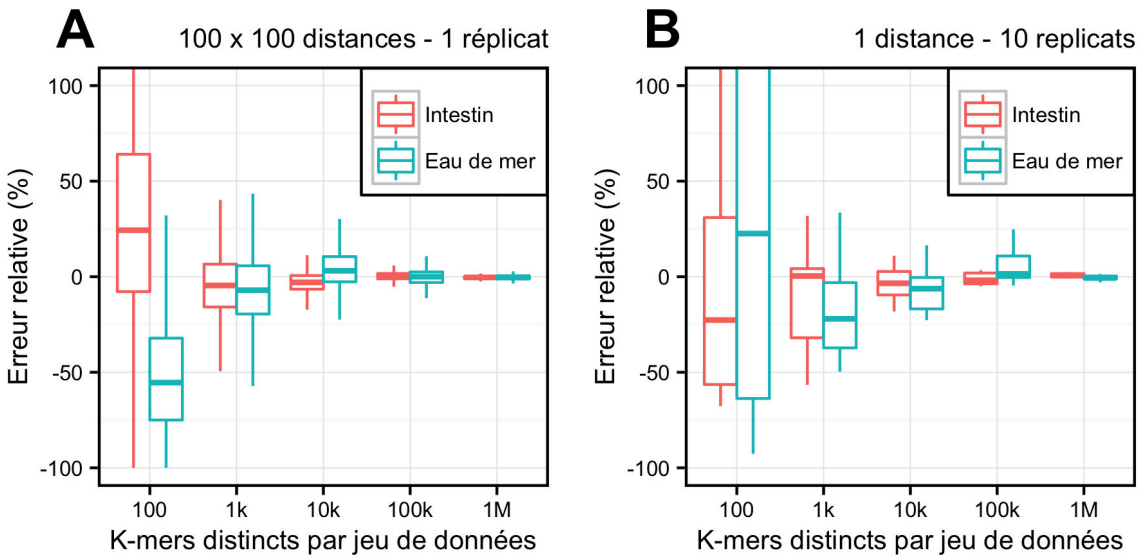

FIGURE 5.8. Erreur d'estimation des distances de Bray-Curtis de SimkaMin. L'erreur est calculée en fonction des distances attendues de SIMKA qui utilisent 500 millions de k-mers distincts par jeu intestinal et 3 milliards par jeu océanique. (**A**) Erreurs d'estimation de 100 × 100 distances en utilisant un seul réplicat. (**B**) Erreur d'estimation d'une seule distance dans 10 réplicats.

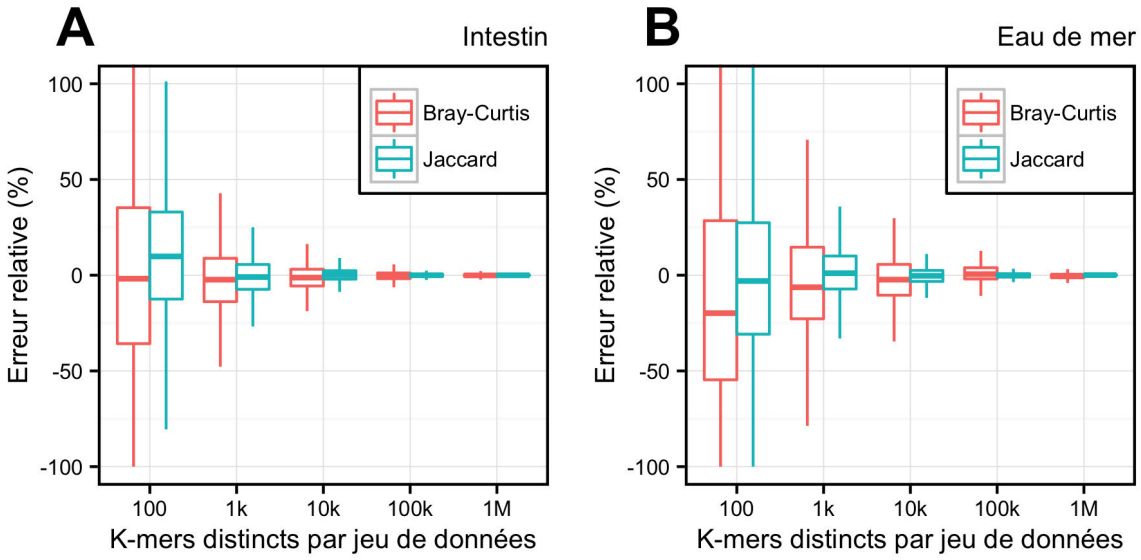

FIGURE 5.9. Erreur d'estimation de la distance de Bray-Curtis de SimkaMin et de la distance de Jaccard de MASH. L'estimation de la distance de Jaccard est fournie par MASH par le biais de la méthode MINHASH. Les boîtes à moustaches représentent les distributions d'erreurs de 10 réplicats contenant chacun 100 × 100 distances.

remarquer que la distance filtrée est légèrement moins biaisée sur les deux projets considérés. Cette différence devient négligeable à partir de 100k k-mers distincts sélectionnés par jeu de données.

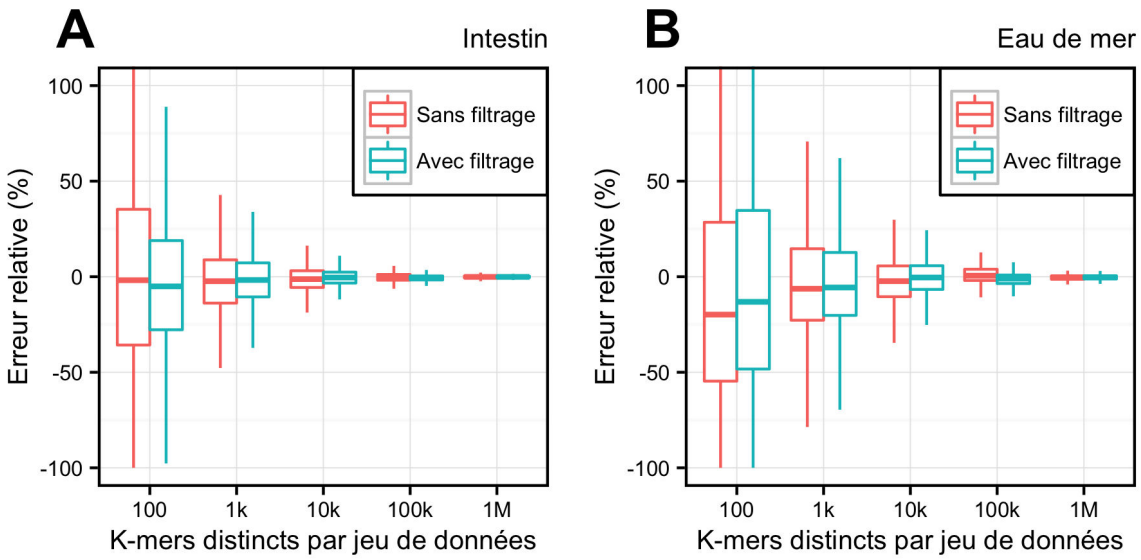

Figure 5.10. Impact du filtre d'abondance sur l'estimation de la distance de Bray-Curtis de SimkaMin. Erreur d'estimation de la distance de Bray-Curtis de SimkaMin filtrée et non filtrée sur deux projets. La distance filtrée ne considère que les k-mers vus au moins deux fois. Les boîtes à moustaches représentent les distributions d'erreurs de 10 réplicats contenant chacun 100×100 distances.

5.4.3 Performances de SimkaMin

Le temps de calcul de SimkaMin a été comparé à celui de Simka et Mash sur 138 jeux de données intestinaux du projet HMP. L'empreinte mémoire et l'utilisation du disque de SimkaMin sont sensiblement les mêmes que celles de Mash (voir section 3.5.1). La figure 5.11 montre que SimkaMin est 4 fois plus rapide que Mash et 8 fois plus rapide que Simka (14 fois plus rapide si le filtre d'abondance est désactivé). Bien sûr, plus les données sont volumineuses plus la différence avec Simka va s'accentuer. En revanche, nous avons trouvé que SimkaMin est toujours 4 fois plus rapide que Mash. Cette différence provient en partie du hachage des k-mers. La fonction de hachage utilisée par SimkaMin et Mash à une complexité linéaire sur la taille de l'élément à hacher (en nombre d'octets). Dans Mash, les k-mers sont traités comme des séquences de k caractères (donc k octets), alors que dans SimkaMin, ils sont compactés via leur représentation en 2 bits par nucléotides (8 octets pour un k-mer de taille 31).

Contrairement à Simka, le filtre d'abondance ne peut qu'augmenter le temps de calcul de Mash et SimkaMin car il génère des accès à un filtre de Bloom. De manière intéressante, on peut voir que ces opérations ont un impact négligeable sur le temps de traitement (figure 5.11).

5.4.4 Conclusion

Dans cette section, nous avons montré empiriquement que la distance de Bray-Curtis de Simka s'estime avec une grande précision en sélectionnant quelques centaines de milliers de k-mers distincts et leur abondance. À partir d'un million d'éléments sélectionnés aléatoirement, il n'y a quasiment plus de biais d'estimation sur les deux projets considérés. L'avantage d'avoir adapté la méthode de Mash est que l'on peut calculer l'estimation de l'index de Jaccard en même temps que celle de Bray-Curtis. L'adaptation que nous proposons n'a aucun impact significatif sur les performances par rapport à la méthode originale de Mash. L'implémentation de SimkaMin est même plus rapide que Mash d'un facteur 4. Ce gain n'est pas négligeable

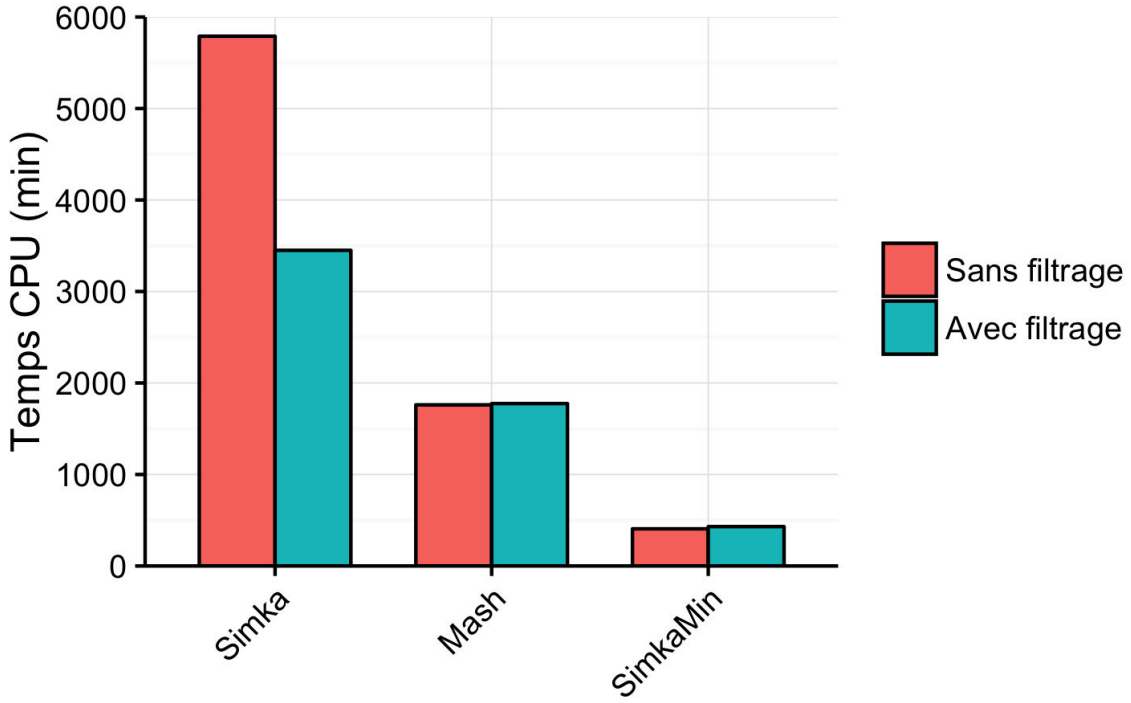

Figure 5.11. Temps CPU de Simka, Mash et SimkaMin. Les tests ont été effectués sur 138 jeux de données intestinaux du projet HMP. Les outils ont été lancés avec $k = 31$, avec et sans filtrage des k-mers vus une seule fois. SimkaMin et Mash ont utilisé 100k k-mers distincts par jeu. La machine de test est équipée d'un processeur Intel E5-2640 de 20 cœurs (2.50 GHz) et 264 Go de mémoire.

compte tenu du temps de traitement important des projets de grande envergure. SimkaMin est donc un outil rapide et complet d'un point de vue distances écologiques.

5.5 Conclusion

Nous avons évalué deux types de sous-échantillonnage : au niveau des lectures et au niveau des k-mers. Dans le premier cas, les estimations observées convergent lentement vers les distances attendues de Simka. Nous déconseillons d'utiliser ce procédé dans l'objectif d'améliorer les performances de Simka. Dans le deuxième cas, nous avons montré empiriquement que la distance quantitative de Bray-Curtis converge très rapidement vers une estimation fiable. Cette seconde étude a abouti à un outil, nommé SimkaMin, qui peut estimer rapidement la distance de Bray-Curtis et la distance de Jaccard. Dans le futur, nous espérons trouver une expression théorique de l'erreur d'estimation de la distance de Bray-Curtis de SimkaMin en fonction de la fraction de k-mers considérés.

Dans l'étude de sous-échantillonnage au niveau des lectures, nous avons vu que les distances quantitatives sont plus robustes au nombre de lectures utilisées. Cette propriété est intéressante car les jeux de données complets sont eux-mêmes des échantillons de la communauté en présence. On peut espérer que les résultats issus de ce type de distances sont reproductibles sur différents réplicats des jeux de données. Dans le cas d'une distance qualitative, nous avons identifié un projet où la distance n'a pas du tout été robuste au sous-échantillonnage (le projet intestinal). Cette famille de distances a le défaut d'être sensible selon l'effort de séquençage produit. Lorsqu'on s'intéresse à la classification des jeux de données, nous avons vu que des groupes d'échantillons

sont localement bien conservés quelle que soit la profondeur de sous-échantillonnage et le type de distances utilisés. Plutôt que de chercher à améliorer les performances de SIMKA, il pourrait être intéressant d'effectuer davantage de calculs dans le but d'améliorer la robustesse des distances. Par exemple, une approche par *bootstrapping* ajoute une valeur de support à chaque branche du clustering hiérarchique. Cependant, elle requiert de nombreuses passes (possiblement des milliers) sur les données en ré-échantillonnant les lectures à chaque itération. Mais nous disposons maintenant de SIMKAMIN. Grâce à ses empreintes mémoire et disque faibles, on pourrait envisager de calculer les nombreux bootstraps en une (ou quelques) passes sur les données. Cette perspective est discutée en section 6.2.2.

Chapitre 6

Conclusion et perspectives

Les travaux de cette thèse apportent quatre contributions aux domaines de la bioinformatique et de la métagénomique : (1) une nouvelle stratégie pour compter simultanément les k-mers de plusieurs jeux de données, nommée *Multiset K-mer Counting* (MKC) ; (2) l'outil de comparaison de métagénomes Simka basé sur le MKC ; (3) l'évaluation en profondeur des distances de Simka basées sur des comptages exacts de k-mers ; et (4) l'outil SimkaMin qui estime rapidement la distance quantitative de Bray-Curtis.

Le MKC est, à notre connaissance, la première stratégie qui compte simultanément les k-mers de nombreux jeux de données avec de hautes performances en termes de temps et de mémoire. La nouveauté de cette stratégie est qu'elle représente les résultats comme un flux de données, fournissant les comptages dans tous les jeux de données, k-mer par k-mer. De plus, celle-ci s'appuie sur les algorithmes de comptage de k-mers de l'état de l'art et pourra continuer de bénéficier de leurs futures améliorations. Le MKC sort du champ de la métagénomique car il existe, aujourd'hui, de nombreuses applications basées sur les k-mers qui considèrent plus d'un jeu de données génomiques. Par exemple, l'outil Cortex [145] améliore la détection de variants génétiques en considérant une population d'individus. L'information génomique de ces individus est représentée par un graphe de *de-Bruijn* coloré. Les nœuds du graphe (c'est-à-dire les k-mers) sont associés à une couleur correspondant au jeu de données d'où ils proviennent. Cette information peut être fournie efficacement par le MKC. De plus, le fait d'avoir accès à tous les comptages d'un k-mer au même instant permet d'appliquer des filtres d'abondance plus complexes qu'un simple seuil.

Simka s'appuie sur le MKC pour calculer une collection de distances écologiques entre les métagénomes. L'intérêt de cette approche est qu'elle s'abstrait d'une phase difficile d'estimation de la diversité taxonomique à partir de données métagénomiques. Celle-ci est remplacée par la diversité en k-mers. Simka est flexible. Les différents paramètres, tels que la taille des k-mers, ont peu d'influence sur ses performances globales. De plus, contrairement à la plupart des outils de l'état de l'art, Simka passe à l'échelle sans la nécessité de filtrer les k-mers vus une seule fois. Dans certains environnements comme le sol, ce filtre est indésirable car la couverture des espèces est globalement très faible. Il existe de nombreuses distances écologiques qui peuvent aboutir à des résultats différents. Nous avons rassemblé un ensemble de distances qui ont la propriété d'être additives sur les k-mers et qui sont donc toutes calculées simultanément en une passe sur les jeux de données.

Une partie de cette thèse a été vouée à l'évaluation des distances basées sur les k-mers. Cela a été d'une importance primordiale car les biologistes préféreraient idéalement travailler sur des comptages d'espèces. Sur le microbiome humain, nous avons trouvé de bonnes relations entre les distances quantitatives taxonomiques et les distances de Simka. Cependant, il n'est pas possible de reproduire une telle expérience sur des environnements plus complexes qui manquent

de bonnes références. L'étude comparative des données de Tara Oceans nous a permis de pousser cette évaluation plus loin. Elle montre notamment que les distances qualitatives de Simka sont cohérentes avec des distances basées sur des OTUs. Cela vient appuyer l'intérêt des spectres de k-mers comme remplacement à une estimation de la diversité taxonomique. L'évaluation a majoritairement porté sur des jeux de données bactériens. Dans le futur, il sera important d'évaluer Simka sur des projets s'intéressant à d'autres types de micro-organismes, comme des eukaryotes, qui sont pour le moment moins étudiés.

Bien que Simka soit un outil performant, sa consommation en disque est problématique. Simka génère des fichiers temporaires de l'ordre de la taille des données à traiter. Pour améliorer les performances globales, nous avons exploré des approches de sous-échantillonnage. L'outil Mash nous a particulièrement inspiré. Celui-ci estime l'index qualitatif de Jaccard en n'utilisant que quelques milliers de k-mers. Nous avons adapté sa méthodologie pour qu'elle prenne en compte l'abondance des k-mers. Nous avons empiriquement mis en évidence que la distance quantitative de Bray-Curtis peut être estimée très précisément en utilisant environ 1 million de k-mers distincts par jeu. Cela représente moins d'1% des k-mers distincts des jeux de données sur les deux projets métagénomiques considérés. Cette adaptation de Mash a été implémentée dans un outil nommé SimkaMin. Celui-ci est 8 fois plus rapide que Simka et 4 fois plus rapide que Mash. Nous espérons prochainement déterminer l'erreur théorique de cette estimation de la distance de Bray-Curtis en fonction du nombre de k-mers utilisés.

6.1 Impact de Simka dans la communauté scientifique

Simka a pour la première fois été utilisé pour comparer les données de Tara Oceans. Cette étude, dans laquelle nous sommes impliqués, a établi la première biogéographie à partir de données métagénomiques. Elle montre que l'organisation génomique des communautés planctoniques est dynamique. Elle évolue différemment selon la taille des organismes. Les courants marins sont le premier facteur à avoir une influence sur la dynamique d'organisation. Au delà d'un an et demi de circulation, l'impact des courants est éclipsé par les variations environnementales. L'article concernant cette étude est actuellement en révision dans le journal *Nature*.

Nous avons référencé d'autres utilisateurs de Simka. Deux études ont abouti à une publication. Dans la première, Dickson *et al.* [146] ont montré que l'exposition des larves de moustiques à différentes bactéries entraîne des variations dans les traits des adultes. Une classification des échantillons issue des résultats de Simka a été déterminée pour confirmer une classification basée sur des OTUs. Les auteurs ont montré un intérêt pour Simka car ses résultats sont indépendants d'une phase d'estimation de la diversité taxonomique. Dans la deuxième étude, Danovaro *et al.* [147] ont exploré un milieu sous-marin ayant subi une éruption volcanique. Simka a été utilisé pour dresser une vue globale de l'organisation des communautés présentes plus ou moins loin du site volcanique.

Pendant un séjour à l'EBI, Simka a été introduit aux utilisateurs de leur pipeline d'analyses métagénomiques [41]. Camilla Speller de l'université de York (BioArch, Department of Archaeology) a proposé de lancer Simka sur des échantillons dentaires archéologiques. Les échantillons ont été prélevés à partir de différents squelettes à plusieurs endroits sur la dent (os, dentine, etc.). Elle nous a montré que le premier critère qui sépare les échantillons est l'endroit sur la dent. Ensuite, de manière intéressante, Simka a séparé les échantillons provenant de squelettes de sites archéologiques différents (anglais et allemand). Enfin, pour un site archéologique donné, Simka est parvenu à regrouper les échantillons par squelette.

Ces différentes applications sont très excitantes. L'EBI a également montré un engouement par rapport à Simka. Dans le futur, nous espérons intégrer l'estimateur SimkaMin dans leur pipeline EMG [41] pour proposer aux utilisateurs de comparer leurs jeux de données en ligne.

6.2 Perspectives

6.2.1 Amélioration de la sensibilité des distances basées sur les *k*-mers

Depuis quelques années, les k-mers sont régulièrement remplacés par les graines espacées. Les graines espacées peuvent être vues comme des k-mers non contigus. Elles autorisent ainsi un certain nombre de substitutions lors de leur comparaison. On les retrouvent dans des applications impliquant des comparaisons de séquences comme : l'alignement de lectures [148], l'alignement de séquences multiples [149], la classification de protéines [150] et la reconstruction phylogénétique [151]. Plus récemment, les graines espacées ont été appliquées aux données métagénomiques. Par exemple, Břinda *et al.* [152] ont montré qu'elles améliorent les approches d'assignation taxonomique sans-alignement, telles que Kraken [37].

Dans le cadre de Simka, les graines espacées permettraient de comparer les métagénomes de manière moins stringente. Les distances seraient alors plus souples et moins sensibles aux erreurs de séquençage et aux variants génomiques. Cette sensibilité est importante dans certaines circonstances. Notamment, les virus sont connus pour avoir des génomes extrêmement mutés.

Le calcul de ces graines a un impact considérable sur le temps de calcul d'un facteur 3 à 5 [152]. Des études s'attaquent actuellement à ce problème [153].

6.2.2 Amélioration de la robustesse des distances

Nos travaux sur le sous-échantillonnage au niveau des lectures indiquent que le clustering des échantillons est sensible au nombre de lectures considérées. Il serait intéressant d'ajouter des valeurs de supports au clustering issue des distances de Simka.

Plusieurs techniques statistiques sont classiquement employées pour ajouter des informations de robustesse à des classifications hiérarchiques. Par exemple, le *bootstrapping* consiste à calculer de nombreux résultats, appelés bootstraps, en sélectionnant aléatoirement les éléments des données originales avec remise. La valeur de bootstrap d'un nœud du clustering original est la fréquence d'apparition de ses bipartitions dans les classifications issues des bootstraps [154]. La technique *Multiscale bootstrap resampling* [155] génère des valeurs de support moins biaisées mais requiert plus de calculs. Elle agglomère différentes valeurs de bootstrap calculées à différentes profondeurs de sous-échantillonnage. Avec Mash et SimkaMin, nous pouvons aujourd'hui imaginer calculer ces nombreux bootstraps efficacement, afin de tirer parti de ces techniques statistiques.

6.2.3 Séquences similaires entre les jeux de données

Déterminer les séquences similaires entre deux ou plusieurs jeux de données a de nombreuses applications. Par exemple, dans le cadre de l'étude du projet Tara Oceans (présentée en section 4.2), cette opération a permis de détecter les lectures spécifiques aux différentes génocénoses. Des analyses taxonomiques (potentiellement coûteuses en ressources informatiques) ont essentiellement été conduites sur ces sous-ensembles de séquences. Une autre application est l'assemblage croisé. L'idée est d'isoler les séquences métagénomiques d'un organisme pour aider à l'assemblage de son génome. On recherche alors les séquences similaires entre plusieurs jeux de données susceptibles de contenir l'organisme cible. Les organismes qui ne sont pas présents dans tous les jeux de données sont automatiquement filtrés.

En s'appuyant sur le MKC, un outil performant pourrait être développé pour sortir ces séquences similaires. En effet, le MKC détermine les k-mers partagés entre les jeux. Seuls ces k-mers peuvent être indexés afin d'identifier rapidement les lectures similaires entre les jeux sans-alignement. De plus, le fait que le MKC fournit les N comptages d'un k-mer permet de construire différentes formules de sélection des k-mers. Par exemple, n'indexer que les k-mers

présents dans ces 3 jeux mais absents dans les 5 autres, etc. Avec en plus la possibilité de prendre en compte l'abondance des k-mers dans les formules.

6.2.4 Mesure de complexité intra-échantillon

En 1960, Whittaker [156] a introduit la notion de diversité-beta pour comparer les communautés sur la base de leurs diversités en espèces. La diversité-beta est intimement liée à la diversité-alpha qui représente la complexité d'un échantillon (le nombre d'espèces dans une communauté). Néanmoins, ces deux notions sont très différentes. Les distances écologiques (diversité-beta) indiquent des changements dans la composition des communautés mais ne donnent pas d'informations par rapport à leur complexité. Par exemple, deux environnements peuvent être dits parfaitement similaires s'ils possèdent une espèce et qu'elle est partagée, ou s'ils possèdent un million d'espèces et qu'elles sont partagées. Cette information de complexité est pertinente et pourrait compléter les mesures de comparaisons de Simka. Bien sûr, il existe des méthodes dédiées pour estimer la diversité, mais c'est encore aujourd'hui un problème ouvert, notamment à partir de données plein-génomes.

Plusieurs observations laissent penser que les spectres de k-mers cachent des informations de complexité : Simka utilise les comptages de k-mers comme un remplacement de la composition taxonomique ; d'autres types d'outils, comme les assembleurs métagénomiques, exploitent les spectres pour détecter des pics d'abondance d'espèces isolées ; nous avons vu que le nombre de k-mers distincts varie en fonction de l'environnement et du type d'organismes étudiés (figure 3.1) ; enfin, en génomique, les spectres de k-mers permettent d'estimer des caractéristiques du génome étudié comme sa taille et son taux d'hétérozygotie [157]. Les spectres de k-mers calculés à partir d'échantillons provenant de milieux complexes sont très bruités mais croiser différents spectres pourrait dévoiler de nouvelles informations. Par exemple, un spectre peut être nettoyé en retirant tous les k-mers partagés avec d'autres jeux. On ne s'intéresse alors plus qu'aux k-mers spécifiques du jeu. À l'inverse, on peut étudier le spectre de k-mers partagés par un certain nombre d'environnements. La mesure de complexité extraite à partir d'un spectre ne pourrait pas être utilisée seule en tant que telle, tout comme les distances de Simka. Cependant, on peut imaginer l'utiliser de manière relative, pour classer les jeux de données par complexité.

6.2.5 Requêtage de jeux de données métagénomiques

Les plateformes d'analyses métagénomiques internationales, telles que MG-RAST [96] et EMG, hébergent actuellement un nombre de jeux de données de l'ordre de la centaine de milliers. Dans le futur, les technologies de séquençage vont continuer à se développer et les projets métagénomiques se multiplier. Ces serveurs atteindront prochainement un nombre de jeux de l'ordre du million. Il est important de bien gérer ces données pour y mener efficacement des opérations complexes. Par exemple, un utilisateur pourrait vouloir comparer ses jeux fraichement séquencés à ceux de projets existants du même environnement. Il pourrait vouloir savoir si une séquence particulière (un gène par exemple) apparait dans les jeux archivés. La gestion des données peut également aider les serveurs eux-mêmes. Par exemple, on peut estimer la composition taxonomique et fonctionnelle d'un nouveau jeu en analysant les jeux (déjà annotés) auquel il est très similaire. Si un nouveau jeu est très similaire aux jeux intestinaux, alors il est inutile d'aligner un catalogue de gènes marins sur celui-ci, etc.

Pour organiser les données, ces serveurs possèdent des systèmes basés sur les annotations des jeux de données (metadonnées), comme leur projet métagénomique, leur environnement, etc. Cependant, ces annotations ne sont pas toujours disponibles, peuvent avoir été perdues ou être difficilement accessibles. De plus, elles peuvent être imprécises. Par exemple, sur le serveur EMG, un jeu de données peut être labellisé "humain" alors qu'il provient plus précisément de

l'intestin. Nous avons réfléchi à deux systèmes de requêtage, basés sur les résultats de Simka, qui pourraient aider à la gestion et à la fouille automatique de données métagénomiques : le requêtage d'un jeu de données contre N jeux et le requêtage d'une séquence particulière contre N jeux.

Requêtage jeu contre jeux. Pour un jeu requête donné, l'objectif est de fournir une liste de jeux ordonnés par similarité vis à vis du jeu requête. Ce type de requêtage est une application directe de Simka. Pendant un séjour à l'EBI, nous avons testé une méthode pour répondre instantanément à ce problème. L'idée est que la similarité entre chaque jeu de données du serveur est pré-calculée par Simka, puis stockée dans une matrice de distances contenant tous les jeux. L'utilisateur choisit son jeu requête sur le serveur. La ligne (ou colonne) correspondante à ce jeu dans la matrice de distances est extraite, puis triée par similarité. Il ne reste alors plus qu'à fournir cette liste à l'utilisateur.

Nous avons pu évaluer la pertinence de ce classement grâce à des mesures du domaine de la fouille de données sur une dizaine de milliers de jeux de données provenant de projets hétérogènes. Nous avons trouvé que les premiers jeux de données (les plus similaires vis à vis d'un jeu requête) proviennent généralement du même projet métagénomique que celui du jeu requête. C'est intéressant d'un point de vue biologique mais il est possible que ce résultat soit biaisé. Les différents projets peuvent utiliser des protocoles d'extraction de l'ADN et de séquençage bien spécifiques. Néanmoins, nous avons découvert que les jeux les plus similaires proviennent généralement du même environnement que la requête. Cela montre que la fouille de données basée sur le contenu génomique est pertinente. Il serait intéressant de continuer ce travail pour trouver dans quels cas la recherche est pertinente ou non, et pourquoi, et quelles distances fonctionnent le mieux. Peut être que certaines distances vont capturer différentes caractéristiques des jeux de données (environnement, saison particulière, température, etc.).

Requêtage séquence contre jeux. Pour une séquence requête donnée, il s'agit de fournir le plus rapidement possible la liste des jeux de données dans laquelle elle apparait. Cette séquence peut être un gène, un génome ou une lecture quelconque. La base de données peut être un projet métagénomique complet, ou encore l'intégralité des données d'un serveur. Bien sûr, le nombre de séquences requêtes peut être immense. Cette application est un des verrous actuels de la bioinformatique des séquences. Elle touche plus au domaine de comparaison de séquences que celui de Simka. Cependant, Simka peut être utilisé pour structurer la base de données.

Actuellement, la comparaison de séquences sans-alignement est employée pour traiter efficacement ce problème. Une séquence requête apparait dans un jeu si elle partage un certain pourcentage de k-mers avec celui-ci. La structure la plus utilisée pour indexer un ensemble de jeux de données se base sur un arbre de filtres de Bloom [158, 159, 160]. Mais ces approches ne sont pas dédiées à la métagénomique et ne passent pas à l'échelle en termes d'espace mémoire sur des projets de grande envergure. Notre idée s'inspire de ces méthodes. Elle commence par le clustering hiérarchique des échantillons de la base de données avec Simka. L'arbre est utilisé pour factoriser le contenu génomique partagé entre les échantillons. Par exemple, si un k-mer apparait dans 4 jeux, il est stocké une seule fois dans leur ancêtre commun le plus proche (LCA). La structure d'indexation de la base de données est une table de hachage associant les k-mers à leur LCA. Pendant cette thèse, de nouvelles structures de données compactes ont été publiées pour indexer de très grands ensembles de k-mers et leur associer des informations [161, 162]. Avec cette méthode, on peut imaginer indexer un projet métagénomique, tel que Tara Oceans, sur une machine possédant environ 500 Go de mémoire vive.

L'approche que nous proposons capture la redondance de la base de données. En revanche, un problème à résoudre pour indexer une quantité plus importante de données est la réduction du

nombre de k-mers considérés. Ce problème fait potentiellement appel à tous les types d'analyses métagénomiques *de novo* pour détecter des k-mers exceptionnels au sein des génomes présents.

Annexes

Programme	Version	Téléchargement
Simka	1.3	https ://gatb.inria.fr/software/simka/
Commet	1	https ://github.com/pierrepeterlongo/commet
Metafast	0.1.0	https ://github.com/ctlab/metafast
Mash	1.1	https ://github.com/marbl/mash

Programme	Ligne de commande
Simka	./simka -in input.txt -out-tmp out-dir -nb-cores 40 -max-memory 100000 -max-count 20 -max-merge 30
Commet	python Commet.py input.txt
Metafast	./metafast.sh -k 31 -b 1 -w out-dir -i input
Mash	./mash sketch -p 40 -r -b 5G -k 31 -s 10000 -o outFilename input*

Table 1. Description des programmes et des lignes de commande utilisées.

Bibliographie

[1] Human Microbiome Project Consortium. Structure, function and diversity of the healthy human microbiome. Nature, 486(7402) :207–214, 2012.

[2] S Dusko Ehrlich et al. Metagenomics of the intestinal microbiota : potential applications. Gastroenterologie clinique et biologique, 34 :S23–S28, 2010.

[3] Eric Karsenti, Silvia G Acinas, Peer Bork, Chris Bowler, Colomban De Vargas, Jeroen Raes, Matthew Sullivan, Detlev Arendt, Francesca Benzoni, Jean-Michel Claverie, et al. A holistic approach to marine eco-systems biology. PLoS biology, 9(10) :e1001177, 2011.

[4] Frederick Sanger, Steven Nicklen, and Alan R Coulson. Dna sequencing with chain-terminating inhibitors. Proceedings of the national academy of sciences, 74(12) :5463–5467, 1977.

[5] Allan M Maxam and Walter Gilbert. A new method for sequencing dna. Proceedings of the National Academy of Sciences, 74(2) :560–564, 1977.

[6] Carl R Woese and George E Fox. Phylogenetic structure of the prokaryotic domain : the primary kingdoms. Proceedings of the National Academy of Sciences, 74(11) :5088–5090, 1977.

[7] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology, 48(3) :443–453, 1970.

[8] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. Journal of molecular biology, 215(3) :403–410, 1990.

[9] Frederick R Blattner, Guy Plunkett, Craig A Bloch, Nicole T Perna, Valerie Burland, Monica Riley, Julio Collado-Vides, Jeremy D Glasner, Christopher K Rode, George F Mayhew, et al. The complete genome sequence of escherichia coli k-12. science, 277(5331) :1453–1462, 1997.

[10] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, et al. The sequence of the human genome. science, 291(5507) :1304–1351, 2001.

[11] Jorge S Reis-Filho. Next-generation sequencing. Breast Cancer Research, 11(3) :S12, 2009.

[12] Jared T Simpson and Richard Durbin. Efficient de novo assembly of large genomes using compressed data structures. Genome research, 22(3) :549–556, 2012.

[13] Daniel R Zerbino and Ewan Birney. Velvet : algorithms for de novo short read assembly using de bruijn graphs. Genome research, 18(5) :821–829, 2008.

[14] Jared T Simpson, Kim Wong, Shaun D Jackman, Jacqueline E Schein, Steven JM Jones, and Inanç Birol. Abyss : a parallel assembler for short read sequence data. Genome research, 19(6) :1117–1123, 2009.

[15] Shaun D Jackman, Benjamin P Vandervalk, Hamid Mohamadi, Justin Chu, Sarah Yeo, S Austin Hammond, Golnaz Jahesh, Hamza Khan, Lauren Coombe, Rene L Warren, et al. Abyss 2.0 : resource-efficient assembly of large genomes using a bloom filter. Genome research, 27(5) :768–777, 2017.

[16] Christoph Bleidorn. Third generation sequencing : technology and its potential impact on evolutionary biodiversity research. Systematics and biodiversity, 14(1) :1–8, 2016.

[17] James T Staley and Allan Konopka. Measurement of in situ activities of nonphotosynthetic microorganisms in aquatic and terrestrial habitats. Annual Reviews in Microbiology, 39(1) :321–346, 1985.

[18] Rudolf I Amann, Wolfgang Ludwig, and Karl-Heinz Schleifer. Phylogenetic identification and in situ detection of individual microbial cells without cultivation. Microbiological reviews, 59(1) :143–169, 1995.

[19] Jared R Leadbetter. Cultivation of recalcitrant microbes : cells are alive, well and revealing their secrets in the 21st century laboratory. Current opinion in microbiology, 6(3) :274–281, 2003.

[20] Baker M. Method offers dna blueprint of a single human cell. Nature, 2012.

[21] Charles Gawad, Winston Koh, and Stephen R Quake. Single-cell genome sequencing : current state of the science. Nature reviews. Genetics, 17(3) :175, 2016.

[22] Jo Handelsman, Michelle R Rondon, Sean F Brady, Jon Clardy, and Robert M Goodman. Molecular biological access to the chemistry of unknown soil microbes : a new frontier for natural products. Chemistry & biology, 5(10) :R245–R249, 1998.

[23] Jack A Gilbert and Christopher L Dupont. Microbial metagenomics : beyond the genome. Annual Review of Marine Science, 3 :347–371, 2011.

[24] Junjie Qin, Ruiqiang Li, Jeroen Raes, Manimozhiyan Arumugam, Kristoffer Solvsten Burgdorf, Chaysavanh Manichanh, Trine Nielsen, Nicolas Pons, Florence Levenez, Takuji Yamada, et al. A human gut microbial gene catalog established by metagenomic sequencing. nature, 464(7285) :59, 2010.

[25] Jeremy A Frank and Søren J Sørensen. Quantitative metagenomic analyses based on average genome size normalization. Applied and environmental microbiology, 77(7) :2513–2521, 2011.

[26] Norman R Pace, David A Stahl, David J Lane, and Gary J Olsen. The analysis of natural microbial populations by ribosomal rna sequences. In Advances in microbial ecology, pages 1–55. Springer, 1986.

[27] Narayan Desai, Dion Antonopoulos, Jack A Gilbert, Elizabeth M Glass, and Folker Meyer. From genomics to metagenomics. Current opinion in biotechnology, 23(1) :72–76, 2012.

[28] Kim D Pruitt, Tatiana Tatusova, and Donna R Maglott. Ncbi reference sequences (refseq) : a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic acids research, 35(suppl_1) :D61–D65, 2006.

[29] James R Cole, Qiong Wang, E Cardenas, J Fish, Benli Chai, Ryan J Farris, AS Kulam-Syed-Mohideen, Donna M McGarrell, T Marsh, George M Garrity, et al. The ribosomal database project : improved alignments and new tools for rrna analysis. Nucleic acids research, 37(suppl_1) :D141–D145, 2008.

[30] Daniel McDonald, Morgan N Price, Julia Goodrich, Eric P Nawrocki, Todd Z DeSantis, Alexander Probst, Gary L Andersen, Rob Knight, and Philip Hugenholtz. An improved greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. The ISME journal, 6(3) :610–618, 2012.

[31] J Gregory Caporaso, Justin Kuczynski, Jesse Stombaugh, Kyle Bittinger, Frederic D Bushman, Elizabeth K Costello, Noah Fierer, Antonio Gonzalez Peña, Julia K Goodrich, Jeffrey I Gordon, et al. Qiime allows analysis of high-throughput community sequencing data. Nature methods, 7(5) :335–336, 2010.

[32] Patrick D Schloss, Sarah L Westcott, Thomas Ryabin, Justine R Hall, Martin Hartmann, Emily B Hollister, Ryan A Lesniewski, Brian B Oakley, Donovan H Parks, Courtney J Robinson, et al. Introducing mothur : open-source, platform-independent, community-supported software for describing and comparing microbial communities. Applied and environmental microbiology, 75(23) :7537–7541, 2009.

[33] Daniel H Huson, Sina Beier, Isabell Flade, Anna Górska, Mohamed El-Hadidi, Suparna Mitra, Hans-Joachim Ruscheweyh, and Rewati Tappu. Megan community edition-interactive exploration and analysis of large-scale microbiome sequencing data. PLoS computational biology, 12(6) :e1004957, 2016.

[34] Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. Bioinformatics, 25(14) :1754–1760, 2009.

[35] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. Nature methods, 9(4) :357–359, 2012.

[36] Susana Vinga and Jonas Almeida. Alignment-free sequence comparison—a review. Bioinformatics, 19(4) :513–523, 2003.

[37] Derrick E Wood and Steven L Salzberg. Kraken : ultrafast metagenomic sequence classification using exact alignments. Genome biology, 15(3) :1, 2014.

[38] Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb Miller. A greedy algorithm for aligning dna sequences. Journal of Computational biology, 7(1-2) :203–214, 2000.

[39] Daehwan Kim, Li Song, Florian P Breitwieser, and Steven L Salzberg. Centrifuge : rapid and sensitive classification of metagenomic sequences. Genome research, 26(12) :1721–1729, 2016.

[40] Peter Menzel, Kim Lee Ng, and Anders Krogh. Fast and sensitive taxonomic classification for metagenomics with kaiju. Nature communications, 7, 2016.

[41] Alex Mitchell, Francois Bucchini, Guy Cochrane, Hubert Denise, Petra ten Hoopen, Matthew Fraser, Sebastien Pesseat, Simon Potter, Maxim Scheremetjew, Peter Sterk, et al. Ebi metagenomics in 2016-an expanding and evolving resource for the analysis and archiving of metagenomic data. Nucleic acids research, 44(D1) :D595–D603, 2015.

[42] Robert D Finn, Jody Clements, and Sean R Eddy. Hmmer web server : interactive sequence similarity searching. Nucleic acids research, 39(suppl_2) :W29–W37, 2011.

[43] Nicola Segata, Levi Waldron, Annalisa Ballarini, Vagheesh Narasimhan, Olivier Jousson, and Curtis Huttenhower. Metagenomic microbial community profiling using unique clade-specific marker genes. Nature methods, 9(8) :811–814, 2012.

[44] Duy Tin Truong, Eric A Franzosa, Timothy L Tickle, Matthias Scholz, George Weingart, Edoardo Pasolli, Adrian Tett, Curtis Huttenhower, and Nicola Segata. Metaphlan2 for enhanced metagenomic taxonomic profiling. Nature methods, 12(10) :902, 2015.

[45] Jonathan A Eisen. Environmental shotgun sequencing : its potential and challenges for studying the hidden world of microbes. PLoS biology, 5(3) :e82, 2007.

[46] Human Microbiome Project Consortium. A framework for human microbiome research. Nature, 486(7402) :215–221, Jun 2012.

[47] Alexander F Koeppel and Martin Wu. Surprisingly extensive mixed phylogenetic and ecological signals among bacterial operational taxonomic units. Nucleic acids research, 41(10) :5175–5188, 2013.

[48] Yunpeng Cai and Yijun Sun. Esprit-tree : hierarchical clustering analysis of millions of 16s rrna pyrosequences in quasilinear computational time. Nucleic acids research, 39(14) :e95–e95, 2011.

[49] Weizhong Li and Adam Godzik. Cd-hit : a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics, 22(13) :1658–1659, 2006.

[50] Robert C Edgar. Search and clustering orders of magnitude faster than blast. Bioinformatics, 26(19) :2460–2461, 2010.

[51] Céline Mercier, Frédéric Boyer, Aurélie Bonin, and E Coissac. Sumatra and sumaclust : fast and exact comparison and clustering of sequences. In Programs and Abstracts of the SeqBio 2013 workshop. Abstract, pages 27–29, 2013.

[52] Frédéric Mahé, Torbjørn Rognes, Christopher Quince, Colomban de Vargas, and Micah Dunthorn. Swarm : robust and fast clustering method for amplicon-based studies. PeerJ, 2 :e593, 2014.

[53] Joseph F Petrosino, Sarah Highlander, Ruth Ann Luna, Richard A Gibbs, and James Versalovic. Metagenomic pyrosequencing and microbial identification. Clinical chemistry, 55(5) :856–866, 2009.

[54] Gwenael Piganeau, Adam Eyre-Walker, Nigel Grimsley, and Herv ? ? Moreau. How and why DNA barcodes underestimate the diversity of microbial eukaryotes. PLoS ONE, 6(2), 2011.

[55] Ufuk Nalbantoglu, Atilla Cakar, Haluk Dogan, Neslihan Abaci, Duran Ustek, Khalid Sayood, and Handan Can. Metagenomic analysis of the microbial community in kefir grains. Food microbiology, 41 :42–51, 2014.

[56] Thomas Vannier, Jade Leconte, Yoann Seeleuthner, Samuel Mondy, Eric Pelletier, Jean-Marc Aury, Colomban De Vargas, Michael Sieracki, Daniele Iudicone, Daniel Vaulot, et al. Survey of the green picoalga bathycoccus genomes in the global ocean. Scientific reports, 6, 2016.

[57] Marcelino T Suzuki and Stephen J Giovannoni. Bias caused by template annealing in the amplification of mixtures of 16s rrna genes by pcr. Applied and environmental microbiology, 62(2) :625–630, 1996.

[58] Silvia G Acinas, Luisa A Marcelino, Vanja Klepac-Ceraj, and Martin F Polz. Divergence and redundancy of 16s rrna sequences in genomes with multiple rrn operons. Journal of bacteriology, 186(9) :2629–2635, 2004.

[59] Evaluation of 16s rDNA-based community profiling for human microbiome research. PLoS ONE, 7(6) :e39315, jun 2012.

[60] Lin Cai, Lin Ye, Amy Hin Yan Tong, Si Lok, and Tong Zhang. Biased diversity metrics revealed by bacterial 16s pyrotags derived from different primer sets. PloS one, 8(1) :e53649, 2013.

[61] John C Wooley, Adam Godzik, and Iddo Friedberg. A primer on metagenomics. PLoS computational biology, 6(2) :e1000667, 2010.

[62] Yu Peng, Henry CM Leung, Siu-Ming Yiu, and Francis YL Chin. Meta-idba : a de novo assembler for metagenomic data. Bioinformatics, 27(13) :i94–i101, 2011.

[63] Nicola Segata, Daniela Boernigen, Timothy L Tickle, Xochitl C Morgan, Wendy S Garrett, and Curtis Huttenhower. Computational meta9omics for microbial community studies. Molecular systems biology, 9(1) :666, 2013.

[64] Toshiaki Namiki, Tsuyoshi Hachiya, Hideaki Tanaka, and Yasubumi Sakakibara. Metavelvet : an extension of velvet assembler to de novo metagenome assembly from short sequence reads. Nucleic acids research, 40(20) :e155–e155, 2012.

[65] Y. Fofanov, Y. Luo, C. Katili, J. Wang, Y. Belosludtsev, T. Powdrill, C. Belapurkar, V. Fofanov, T.-B. Li, S. Chumakov, and B. M. Pettitt. How independent are the appearances of n-mers in different genomes ? Bioinformatics, 20(15) :2421–2428, apr 2004.

[66] Yu-Wei Wu and Yuzhen Ye. A novel abundance-based algorithm for binning metagenomic sequences using l-tuples. Journal of Computational Biology, 18(3) :523–534, 2011.

[67] Carlotta De Filippo, Matteo Ramazzotti, Paolo Fontana, and Duccio Cavalieri. Bioinformatic approaches for functional annotation and pathway inference in metagenomics data. Briefings in bioinformatics, 13(6) :696–710, 2012.

[68] Alexander Sczyrba, Peter Hofmann, Peter Belmann, David Koslicki, Stefan Janssen, Johannes Droege, Ivan Gregor, Stephan Majda, Jessika Fiedler, Eik Dahms, et al. Critical assessment of metagenome interpretation- a benchmark of computational metagenomics software. Biorxiv, page 099127, 2017.

[69] Samuel Kariin and Chris Burge. Dinucleotide relative abundance extremes : a genomic signature. Trends in genetics, 11(7) :283–290, 1995.

[70] Hanno Teeling, Jost Waldmann, Thierry Lombardot, Margarete Bauer, and Frank O Glöckner. Tetra : a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in dna sequences. BMC bioinformatics, 5(1) :163, 2004.

[71] Hanno Teeling and Frank Oliver Glöckner. Current opportunities and challenges in microbial metagenome analysis—a bioinformatic perspective. Briefings in bioinformatics, 13(6) :728–742, 2012.

[72] Yi Wang, Henry CM Leung, Siu-Ming Yiu, and Francis YL Chin. Metacluster 4.0 : a novel binning algorithm for ngs reads and huge number of species. Journal of Computational Biology, 19(2) :241–249, 2012.

[73] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[74] Johannes Alneberg, Brynjar Smári Bjarnason, Ino De Bruijn, Melanie Schirmer, Joshua Quick, Umer Z Ijaz, Leo Lahti, Nicholas J Loman, Anders F Andersson, and Christopher Quince. Binning metagenomic contigs by coverage and composition. Nature methods, 11(11) :1144–1146, 2014.

[75] Dongwan D Kang, Jeff Froula, Rob Egan, and Zhong Wang. Metabat, an efficient tool for accurately reconstructing single genomes from complex microbial communities. PeerJ, 3 :e1165, 2015.

[76] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast : a new generation of protein database search programs. Nucleic acids research, 25(17) :3389–3402, 1997.

[77] Katharina J Hoff, Maike Tech, Thomas Lingner, Rolf Daniel, Burkhard Morgenstern, and Peter Meinicke. Gene prediction in metagenomic fragments : a large scale machine learning approach. BMC bioinformatics, 9(1) :217, 2008.

[78] Katharina J Hoff, Thomas Lingner, Peter Meinicke, and Maike Tech. Orphelia : predicting genes in metagenomic sequencing reads. Nucleic acids research, 37(suppl_2) :W101–W105, 2009.

[79] Lawrence Rabiner and B Juang. An introduction to hidden markov models. ieee assp magazine, 3(1) :4–16, 1986.

[80] Alex Bateman, Lachlan Coin, Richard Durbin, Robert D Finn, Volker Hollich, Sam Griffiths-Jones, Ajay Khanna, Mhairi Marshall, Simon Moxon, Erik LL Sonnhammer, et al. The pfam protein families database. Nucleic acids research, 32(suppl_1) :D138–D141, 2004.

[81] UniProt Consortium et al. Reorganizing the protein space at the universal protein resource (uniprot). Nucleic acids research, page gkr981, 2011.

[82] Robert D Finn, Teresa K Attwood, Patricia C Babbitt, Alex Bateman, Peer Bork, Alan J Bridge, Hsin-Yu Chang, Zsuzsanna Dosztányi, Sara El-Gebali, Matthew Fraser, et al. Interpro in 2017—beyond protein family and domain annotations. Nucleic acids research, 45(D1) :D190–D199, 2016.

[83] Minoru Kanehisa, Susumu Goto, Shuichi Kawashima, Yasushi Okuno, and Masahiro Hattori. The kegg resource for deciphering the genome. Nucleic acids research, 32(suppl_1) :D277–D280, 2004.

[84] Rachel S Poretsky, Nasreen Bano, Alison Buchan, Gary LeCleir, Jutta Kleikemper, Maria Pickering, Whitney M Pate, Mary Ann Moran, and James T Hollibaugh. Analysis of microbial gene transcripts in environmental samples. Applied and Environmental Microbiology, 71(7) :4121–4126, 2005.

[85] Jack A Gilbert, Dawn Field, Ying Huang, Rob Edwards, Weizhong Li, Paul Gilna, and Ian Joint. Detection of large numbers of novel sequences in the metatranscriptomes of complex marine microbial communities. PloS one, 3(8) :e3042, 2008.

[86] Pierre Legendre and Miquel De Cáceres. Beta diversity as the variance of community data : dissimilarity coefficients and partitioning. Ecol Lett, 16(8) :951–963, Aug 2013.

[87] J Roger Bray and John T Curtis. An ordination of the upland forest communities of southern wisconsin. Ecological monographs, 27(4) :325–349, 1957.

[88] Catherine Lozupone and Rob Knight. Unifrac : a new phylogenetic method for comparing microbial communities. Applied and environmental microbiology, 71(12) :8228–8235, 2005.

[89] Manimozhiyan Arumugam, Jeroen Raes, Eric Pelletier, Denis Le Paslier, Takuji Yamada, Daniel R. Mende, Gabriel R. Fernandes, Julien Tap, Thomas Bruls, Jean-Michel Batto, Marcelo Bertalan, Natalia Borruel, Francesc Casellas, Leyden Fernandez, Laurent Gautier, Torben Hansen, Masahira Hattori, Tetsuya Hayashi, Michiel Kleerebezem, Ken Kurokawa, Marion Leclerc, Florence Levenez, Chaysavanh Manichanh, H. Bjorn Nielsen, Trine Nielsen, Nicolas Pons, Julie Poulain, Junjie Qin, Thomas Sicheritz-Ponten, Sebastian Tims, David Torrents, Edgardo Ugarte, Erwin G. Zoetendal, Jun Wang, Francisco Guarner, Oluf Pedersen, Willem M. de Vos, Soren Brunak, Joel Dore, Jean Weissenbach, S. Dusko Ehrlich, and Peer Bork. Enterotypes of the human gut microbiome. Nature, 473(7346) :174–180, 05 2011.

[90] Gary D Wu, Jun Chen, Christian Hoffmann, Kyle Bittinger, Ying-Yu Chen, Sue A Keilbaugh, Meenakshi Bewtra, Dan Knights, William A Walters, Rob Knight, et al. Linking long-term dietary patterns with gut microbial enterotypes. Science, 334(6052) :105–108, 2011.

[91] Susan M Huse, Yuzhen Ye, Yanjiao Zhou, and Anthony A Fodor. A core human microbiome as viewed through 16s rrna sequence clusters. PloS one, 7(6) :e34242, 2012.

[92] Sahar Abubucker, Nicola Segata, Johannes Goll, Alyxandria M Schubert, Jacques Izard, Brandi L Cantarel, Beltran Rodriguez-Mueller, Jeremy Zucker, Mathangi Thiagarajan, Bernard Henrissat, et al. Metabolic reconstruction for metagenomic data and its application to the human microbiome. PLoS computational biology, 8(6) :e1002358, 2012.

[93] Elizabeth A Grice and Julia A Segre. The skin microbiome. Nature reviews. Microbiology, 9(4) :244, 2011.

[94] Susannah G Tringe, Tao Zhang, Xuguo Liu, Yiting Yu, Wah Heng Lee, Jennifer Yap, Fei Yao, Sim Tiow Suan, Seah Keng Ing, Matthew Haynes, et al. The airborne metagenome in an indoor urban environment. PloS one, 3(4) :e1862, 2008.

[95] Simon Lax, Daniel P Smith, Jarrad Hampton-Marcell, Sarah M Owens, Kim M Handley, Nicole M Scott, Sean M Gibbons, Peter Larsen, Benjamin D Shogan, Sophie Weiss, et al. Longitudinal analysis of microbial interaction between humans and the indoor environment. Science, 345(6200) :1048–1052, 2014.

[96] Folker Meyer, Daniel Paarmann, Mark D'Souza, Robert Olson, Elizabeth M Glass, Michael Kubal, Tobias Paczian, A Rodriguez, Rick Stevens, Andreas Wilke, et al. The metagenomics rast server–a public resource for the automatic phylogenetic and functional analysis of metagenomes. BMC bioinformatics, 9(1) :386, 2008.

[97] Tom O Delmont, Patrick Robe, Ian Clark, Pascal Simonet, and Timothy M Vogel. Metagenomic comparison of direct and indirect soil dna extraction approaches. Journal of microbiological methods, 86(3) :397–400, 2011.

[98] Tom O Delmont, Emmanuel Prestat, Kevin P Keegan, Michael Faubladier, Patrick Robe, Ian M Clark, Eric Pelletier, Penny R Hirsch, Folker Meyer, Jack A Gilbert, et al. Structure, fluctuation and magnitude of a natural grassland soil metagenome. The ISME journal, 6(9) :1677, 2012.

[99] Tom O Delmont, Pascal Simonet, and Timothy M Vogel. Describing microbial communities and performing global comparisons in the 'omic era. The ISME journal, 6(9) :1625, 2012.

[100] Douglas B Rusch, Aaron L Halpern, Granger Sutton, Karla B Heidelberg, Shannon Williamson, Shibu Yooseph, Dongying Wu, Jonathan A Eisen, Jeff M Hoffman, Karin Remington, et al. The sorcerer ii global ocean sampling expedition : northwest atlantic through eastern tropical pacific. PLoS biology, 5(3) :e77, 2007.

[101] MetaSUB International Consortium et al. The metagenomics and metadesign of the subways and urban biomes (metasub) international consortium inaugural meeting report. Microbiome, 4(1) :1–14, 2016.

[102] Anna Kopf, Mesude Bicak, Renzo Kottmann, Julia Schnetzer, Ivaylo Kostadinov, Katja Lehmann, Antonio Fernandez-Guerra, Christian Jeanthon, Eyal Rahav, Matthias Ullrich, et al. The ocean sampling day consortium. Gigascience, 4(1) :27, 2015.

[103] W James Kent. Blat—the blast-like alignment tool. Genome research, 12(4) :656–664, 2002.

[104] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. Journal of molecular biology, 147(1) :195–197, 1981.

[105] Danai Fimereli, Vincent Detours, and Tomasz Konopka. Triagetools : tools for partitioning and prioritizing analysis of high-throughput sequencing data. Nucleic acids research, page gkt094, 2013.

[106] Nicolas Maillet, Claire Lemaitre, Rayan Chikhi, Dominique Lavenier, and Pierre Peterlongo. Compareads : comparing huge metagenomic experiments. BMC bioinformatics, 13(19) :S10, 2012.

[107] Nicolas Maillet, Guillaume Collet, Thomas Vannier, Dominique Lavenier, and Pierre Peterlongo. Commet : comparing and combining multiple metagenomic datasets. In Bioinformatics and Biomedicine (BIBM), 2014 IEEE International Conference on, pages 94–98. IEEE, 2014.

[108] Brian D. Ondov, Todd J. Treangen, Páll Melsted, Adam B. Mallonee, Nicholas H. Bergman, Sergey Koren, and Adam M. Phillippy. Mash : fast genome and metagenome distance estimation using minhash. Genome Biol, 17(1) :132, 2016.

[109] Andrei Z Broder. On the resemblance and containment of documents. In Compression and Complexity of Sequences 1997. Proceedings, pages 21–29. IEEE, 1997.

[110] Ruiqiang Li, Wei Fan, Geng Tian, Hongmei Zhu, Lin He, Jing Cai, Quanfei Huang, Qingle Cai, Bo Li, Yinqi Bai, et al. The sequence and de novo assembly of the giant panda genome. Nature, 463(7279) :311, 2010.

[111] Guillaume Marçais and Carl Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics, 27(6) :764–770, 2011.

[112] Guillaume Rizk, Dominique Lavenier, and Rayan Chikhi. Dsk : k-mer counting with very low memory usage. Bioinformatics, page btt020, 2013.

[113] Sebastian Deorowicz, Marek Kokot, Szymon Grabowski, and Agnieszka Debudaj-Grabysz. Kmc 2 : Fast and resource-frugal k-mer counting. Bioinformatics, 31(10) :1569–1576, 2015.

[114] Michael Roberts, Wayne Hayes, Brian R Hunt, Stephen M Mount, and James A Yorke. Reducing storage requirements for biological sequence comparison. Bioinformatics, 20(18) :3363–3369, 2004.

[115] Yang Li et al. Mspkmercounter : a fast and memory efficient approach for k-mer counting. arXiv preprint arXiv :1505.06550, 2015.

[116] Marek Kokot, Sebastian Deorowicz, and Agnieszka Debudaj-Grabysz. Sorting data on ultra-large scale with raduls. In International Conference : Beyond Databases, Architectures and Structures, pages 235–245. Springer, 2017.

[117] Veronika B Dubinkina, Dmitry S Ischenko, Vladimir I Ulyantsev, Alexander V Tyakht, and Dmitry G Alexeev. Assessment of k-mer spectrum applicability for metagenomic dissimilarity analysis. BMC bioinformatics, 17(1) :38, 2016.

[118] Sohan Seth, Niko Välimäki, Samuel Kaski, and Antti Honkela. Exploration and retrieval of whole-metagenome sequencing samples. Bioinformatics, 30(17) :2471–2479, 2014.

[119] Vladimir I. Ulyantsev, Sergey V. Kazakov, Veronika B. Dubinkina, Alexander V. Tyakht, and Dmitry G. Alexeev. Metafast : fast reference-free graph-based comparison of shotgun metagenomic data. Bioinformatics, Jun 2016.

[120] Human Microbiome Project Consortium et al. Structure, function and diversity of the healthy human microbiome. nature, 486(7402) :207, 2012.

[121] Junjie Qin, Yingrui Li, Zhiming Cai, Shenghui Li, Jianfeng Zhu, Fan Zhang, Suisha Liang, Wenwei Zhang, Yuanlin Guan, Dongqian Shen, et al. A metagenome-wide association study of gut microbiota in type 2 diabetes. Nature, 490(7418) :55–60, 2012.

[122] Edward M McCreight. A space-economical suffix tree construction algorithm. Journal of the ACM (JACM), 23(2) :262–272, 1976.

[123] Niko Välimäki and Simon J Puglisi. Distributed string mining for high-throughput sequencing data. In WABI, pages 441–452. Springer, 2012.

[124] Ebrahim Afshinnekoo, Cem Meydan, Shanin Chowdhury, Dyala Jaroudi, Collin Boyer, Nick Bernstein, Julia M Maritz, Darryl Reeves, Jorge Gandara, Sagar Chhangawala, et al. Geospatial resolution of human and bacterial diversity with city-scale metagenomics. Cell systems, 1(1) :72–87, 2015.

[125] Peter Deutsch and Jean-Loup Gailly. Zlib compressed data format specification version 3.3. Technical report, RFC 1950, May, 1996.

[126] Sandrine Pavoine, Errol Vela, Sophie Gachet, Gérard de Bélair, and Michael B. Bonsall. Linking patterns in phylogeny, traits, abiotic variables and space : a novel approach to linking environmental filtering and plant community assembly. Journal of Ecology, 99(1) :165–175, 2011.

[127] Omry Koren, Dan Knights, Antonio Gonzalez, Levi Waldron, Nicola Segata, Rob Knight, Curtis Huttenhower, and Ruth E Ley. A guide to enterotypes across the human body : meta-analysis of microbial community structures in human microbiome datasets. PLoS Comput Biol, 9(1) :e1002863, 2013.

[128] Anne Chao, Robin L. Chazdon, Robert K. Colwell, and Tsung-Jen Shen. Abundance-based similarity indices and their estimation when there are unseen species in samples. Biometrics, 62(2) :361–371, Jun 2006.

[129] Erwan Drezen, Guillaume Rizk, Rayan Chikhi, Charles Deltel, Claire Lemaitre, Pierre Peterlongo, and Dominique Lavenier. Gatb : Genome assembly & analysis tool box. Bioinformatics, 30(20) :2959–2961, 2014.

[130] Veronika B. Dubinkina, Dmitry S. Ischenko, Vladimir I. Ulyantsev, Alexander V. Tyakht, and Dmitry G. Alexeev. Assessment of k-mer spectrum applicability for metagenomic dissimilarity analysis. BMC Bioinformatics, 17 :38, 2016.

[131] I. Borg and P. Groenen. Modern Multidimensional Scaling : Theory and Applications. Springer Series in Statistics. Springer New York, 2013.

[132] Elizabeth K Costello, Christian L Lauber, Micah Hamady, Noah Fierer, Jeffrey I Gordon, and Rob Knight. Bacterial community variation in human body habitats across space and time. Science, 326(5960) :1694–1697, 2009.

[133] Luis M Rodriguez-r and Konstantinos T Konstantinidis. Nonpareil : a redundancy-based approach to assess the level of coverage in metagenomic datasets. Bioinformatics, 30(5) :629–635, 2013.

[134] Robert R Sokal. A statistical method for evaluating systematic relationships. University of Kansas Scientific Bulletin, 38 :1409–1438, 1958.

[135] Jennifer B Hughes Martiny, Brendan JM Bohannan, James H Brown, Robert K Colwell, Jed A Fuhrman, Jessica L Green, M Claire Horner-Devine, Matthew Kane, Jennifer Adams Krumins, Cheryl R Kuske, et al. Microbial biogeography : putting microorganisms on the map. Nature reviews. Microbiology, 4(2) :102, 2006.

[136] China A Hanson, Jed A Fuhrman, M Claire Horner-Devine, and Jennifer BH Martiny. Beyond biogeographic patterns : processes shaping the microbial landscape. Nature reviews. Microbiology, 10(7) :497, 2012.

[137] Jennifer R Brum, J Cesar Ignacio-Espinoza, Simon Roux, Guilhem Doulcier, Silvia G Acinas, Adriana Alberti, Samuel Chaffron, Corinne Cruaud, Colomban De Vargas, Josep M Gasol, et al. Patterns and ecological drivers of ocean viral communities. Science, 348(6237) :1261498, 2015.

[138] Sébastien Boutin, Simon Y. Graeber, Michael Weitnauer, Jessica Panitz, Mirjam Stahl, Diana Clausznitzer, Lars Kaderali, Gisli Einarsson, Michael M. Tunney, J. Stuart Elborn, Marcus A. Mall, and Alexander H. Dalpke. Comparison of microbiomes from different niches of upper and lower airways in children and adolescents with cystic fibrosis. PLoS ONE, 10(1) :1–19, 01 2015.

[139] A. Shade, S. E. Jones, J. G. Caporaso, J. Handelsman, R. Knight, N. Fierer, and J. A. Gilbert. Conditionally rare taxa disproportionately contribute to temporal changes in microbial diversity. mBio, 5(4) :e01371–14–e01371–14, jul 2014.

[140] S. Genitsaris, S. Monchy, E. Viscogliosi, T. Sime-Ngando, S. Ferreira, and U. Christaki. Seasonal variations of marine protist community structure based on taxon-specific traits using the eastern english channel as a model coastal system. FEMS Microbiology Ecology, 91(5) :fiv034–fiv034, mar 2015.

[141] Suzanne Coveley, Mostafa S Elshahed, and Noha H Youssef. Response of the rare biosphere to environmental stressors in a highly diverse ecosystem (zodletone spring, ok, usa). PeerJ, 3 :e1182, 2015.

[142] V. Gomez-Alvarez, S. Pfaller, J. G. Pressman, D. G. Wahman, and R. P. Revetta. Resilience of microbial communities in a simulated drinking water distribution system subjected to disturbances : role of conditionally rare taxa and potential implications for antibiotic-resistant bacteria. Environ. Sci. : Water Res. Technol., 2(4) :645–657, 2016.

[143] David F Robinson and Leslie R Foulds. Comparison of phylogenetic trees. Mathematical biosciences, 53(1-2) :131–147, 1981.

[144] Mary K Kuhner and Joseph Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. Molecular biology and evolution, 11(3) :459–468, 1994.

[145] Zamin Iqbal, Mario Caccamo, Isaac Turner, Paul Flicek, and Gil McVean. De novo assembly and genotyping of variants using colored de bruijn graphs. Nature genetics, 44(2) :226–232, 2012.

[146] Laura B Dickson, Davy Jiolle, Guillaume Minard, Isabelle Moltini-Conclois, Stevenn Volant, Amine Ghozlane, Christiane Bouchier, Diego Ayala, Christophe Paupy, Claire Valiente Moro, et al. Carryover effects of larval exposure to different environmental bacteria drive adult trait variation in a mosquito vector. Science Advances, 3(8) :e1700585, 2017.

[147] Roberto Danovaro, Miquel Canals, Michael Tangherlini, Antonio Dell'Anno, Cristina Gambi, Galderic Lastras, David Amblas, Anna Sanchez-Vidal, Jaime Frigola, Antoni M Calafat, et al. A submarine volcanic eruption leads to a novel microbial habitat. Nature Ecology & Evolution, 1(6) :s41559–017, 2017.

[148] Stephen M Rumble, Phil Lacroute, Adrian V Dalca, Marc Fiume, Arend Sidow, and Michael Brudno. Shrimp : accurate mapping of short color-space reads. PLoS computational biology, 5(5) :e1000386, 2009.

[149] Aaron E Darling, Todd J Treangen, Louxin Zhang, Carla Kuiken, Xavier Messeguer, and Nicole T Perna. Procrastination leads to efficient filtration for local multiple alignment. In International Workshop on Algorithms in Bioinformatics, pages 126–137. Springer, 2006.

[150] Taku Onodera and Tetsuo Shibuya. The gapped spectrum kernel for support vector machines. In International Workshop on Machine Learning and Data Mining in Pattern Recognition, pages 1–15. Springer, 2013.

[151] Chris-Andre Leimeister, Marcus Boden, Sebastian Horwege, Sebastian Lindner, and Burkhard Morgenstern. Fast alignment-free sequence comparison using spaced-word frequencies. Bioinformatics, 30(14) :1991–1999, 2014.

[152] Karel Břinda, Maciej Sykulski, and Gregory Kucherov. Spaced seeds improve k-mer-based metagenomic classification. Bioinformatics, 31(22) :3584–3592, 2015.

[153] Samuele Girotto, Matteo Comin, and Cinzia Pizzi. Fast Spaced Seed Hashing. In Russell Schwartz and Knut Reinert, editors, 17th International Workshop on Algorithms in Bioinformatics (WABI 2017), volume 88 of Leibniz International Proceedings in Informatics (LIPIcs), pages 7 :1–7 :14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[154] Joseph Felsenstein. Confidence limits on phylogenies : an approach using the bootstrap. Evolution, 39(4) :783–791, 1985.

[155] Hidetoshi Shimodaira. An approximately unbiased test of phylogenetic tree selection. Systematic biology, 51(3) :492–508, 2002.

[156] Robert Harding Whittaker. Vegetation of the siskiyou mountains, oregon and california. Ecological monographs, 30(3) :279–338, 1960.

[157] Rayan Chikhi and Paul Medvedev. Informed and automated k-mer size selection for genome assembly. Bioinformatics, 30(1) :31–37, 2013.

[158] Brad Solomon and Carl Kingsford. Fast search of thousands of short-read sequencing experiments. Nature biotechnology, 34(3) :300, 2016.

[159] Chen Sun, Robert S Harris, Rayan Chikhi, and Paul Medvedev. Allsome sequence bloom trees. In International Conference on Research in Computational Molecular Biology, pages 272–286. Springer, 2017.

[160] Brad Solomon and Carl Kingsford. Improved search of large transcriptomic sequencing databases using split sequence bloom trees. In International Conference on Research in Computational Molecular Biology, pages 257–271. Springer, 2017.

[161] Antoine Limasset, Guillaume Rizk, Rayan Chikhi, and Pierre Peterlongo. Fast and scalable minimal perfect hashing for massive key sets. arXiv preprint arXiv :1702.03154, 2017.

[162] Camille Marchet, Lolita Lecompte, Antoine Limasset, Lucie Bittner, and Pierre Peterlongo. A resource-frugal probabilistic dictionary and applications in bioinformatics. arXiv preprint arXiv :1703.00667, 2017.

Publications

Articles

— Multiple comparative metagenomics using multiset k-mer counting
G Benoit, P Peterlongo, M Mariadassou, E Drezen, S Schbath, D Lavenier, C Lemaitre
PeerJ Computer Science, 2016

— Reference-free compression of high throughput sequencing data with a probabilistic de Bruijn graph
G Benoit, C Lemaitre, D Lavenier, E Drezen, T Dayris, R Uricaru, G Rizk
BMC Bioinformatics, 2015

Chapitre de livre

— de novo NGS Data Compression
G Benoit, C Lemaitre, G Rizk, E Drezen, D Lavenier
Algorithms for Next-Generation Sequencing Data : Techniques, Approaches, and Applications, M. Eloumi (editor), Springer, 2017

Présentations

— Simka: large scale de novo comparative metagenomics
G Benoit, P Peterlongo, M Mariadassou, E Drezen, S Schbath, D Lavenier, C Lemaitre
JOBIM, 2017
Communication à un congrès national (sans actes)

— Simka: fast kmer-based method for estimating the similarity between numerous metagenomic datasets
G Benoit, P Peterlongo, D Lavenier, C Lemaitre
RCAM, 2015
Communication à un congrès international (sans actes)

— Simka: fast kmer-based method for estimating the similarity between numerous metagenomic datasets
G Benoit, P Peterlongo, D Lavenier, C Lemaitre
EBAME, 2015
Communication à un congrès national (sans actes)

— Reference-free compression of high throughput sequencing data with a probabilistic de Bruijn graph
G Benoit, C Lemaitre, D Lavenier, E Drezen, T Dayris, R Uricaru, G Rizk
SeqBio, 2015
Communication à un congrès national (sans actes)

Posters

— Simka: fast kmer-based method for estimating the similarity between numerous metagenomic datasets
G Benoit, P Peterlongo, D Lavenier, C Lemaitre
JOBIM, 2015

— Bloocoo, a memory efficient read corrector
G Benoit, D Lavenier, C Lemaitre, G Rizk
ECCB, 2014

Liste des tableaux

3.1 Définition des distances calculées par Simka. 47
3.2 Performances de Simka et statistiques des k-mers sur le projet HMP entier. 52

1 Description des programmes et des lignes de commande utilisées. 99

Liste des Algorithmes

1 Calcul de la distance de Bray-Curtis (équation 3.1) entre N jeux de lectures. . . . 45

2 Sélection aléatoire de n k-mers distincts et leur abondance exacte dans un jeu de données. La liste L est automatiquement triée dés qu'un élément y est inséré (en $O(log\, n)$ opérations). La valeur L_{top} est automatiquement mise à jour en lisant le dernier élément de L. 87

Table des figures

1.1 Représentation d'une classification phylogénétique. 6
1.2 Similarité entre les échantillons en termes de contenu génomique partagé 20

2.1 Aperçu d'une méthode de métagénomique comparative *de novo*. 24
2.2 Processus de comparaison de deux jeux de lectures de TriageTools 27
2.3 Processus de comparaison de MASH en utilisant la technique Minhash 31
2.4 Représentation des k-mers d'une lecture et leur minimizer. 34
2.5 Corrélation entre des distances taxonomiques et des distances *de novo* basées sur des petits k-mers. 36

3.1 Nombre moyen de k-mers distincts par jeu en fonction du type d'environnement et du type d'organismes ($k = 21$). ... 40
3.2 Stratégie de SIMKA ... 41
3.3 Stratégie de comptage de k-mers multi-jeux avec k=3 ... 43
3.4 Représentation des spectres de k-mers dans Simka. ... 49
3.5 Performances de SIMKA et des outils de l'état de l'art par rapport à un nombre N de jeux de lectures à comparer. ... 51
3.6 Impact de la taille des k-mers sur les performances de SIMKA. ... 53
3.7 Scalabilité de SIMKA. ... 54

4.1 Comparaison des mesures de similarité de Simka et Commet. ... 59
4.2 Comparaison des résultats de Simka et de BLAT pour différentes valeurs de k et plusieurs seuils d'identité de BLAT. ... 60
4.3 Pourcentage de lectures alignées sur des références par tissu. ... 61
4.4 Corrélation entre des distances quantitatives taxonomiques et *de novo*. ... 62
4.5 Corrélation entre des distances qualitatives taxonomiques et *de novo*. ... 63
4.6 Impact de la taille des k-mers et du filtre d'abondance sur la corrélation avec des distances taxonomiques. ... 63
4.7 Distribution de la diversité des jeux de lectures du projet HMP par tissus. ... 65
4.8 Abondances relatives des genres principaux des échantillons d'intestin du projet HMP. ... 66
4.9 Résultats de Simka sur des jeux de données peu couverts du projet Global Ocean Sampling (GOS) ... 67
4.10 Partitionnement des échantillons de Tara Oceans en génocénoses. ... 69
4.11 Représentation des stations de Tara Oceans et des génocénoses ... 71
4.12 Corrélation entre des distances basées sur des OTUs et des distances de Simka. ... 72

5.1 Concept de bipartition dans un clustering hiérarchique. ... 77
5.2 Erreur d'estimation des distances issues du sous-échantillonnage au niveau des lectures. ... 79
5.3 Corrélation entre les distances de Simka et les distances issues du sous-échantillonnage au niveau des lectures. ... 80
5.4 Impact du sous-échantillonnage au niveau des lectures sur la classification des jeux de données intestinaux. ... 81
5.5 Impact du sous-échantillonnage au niveau des lectures sur la classification des jeux de données océaniques. ... 82
5.6 Impact du sous-échantillonnage au niveau des lectures sur la classification des jeux de données. ... 83
5.7 Impact du sous-échantillonnage au niveau des vecteurs d'abondances sur la distance de Bray-Curtis. ... 85
5.8 Erreur d'estimation des distances de Bray-Curtis de SimkaMin. ... 89
5.9 Erreur d'estimation de la distance de Bray-Curtis de SimkaMin et de la distance de Jaccard de MASH. ... 89
5.10 Impact du filtre d'abondance sur l'estimation de la distance de Bray-Curtis de SimkaMin. ... 90
5.11 Temps CPU de Simka, Mash et SimkaMin. ... 91